A Prediction Model Based on Ratings and Reviews of Online Commodities

March 9, 2020

Contents

1 Abstract

Keywords: Sentiment analysis, Logistic regression, Time series analysis

1.1 Introduction

Amazon, the largest e-commerce company in the United States, has created a satisfaction degree evaluation system for its customers in the online marketplace, allowing the customers to provide feedbacks to their orders and purchases. Generally, there are two ways to express the satisfaction degrees, which are called "star ratings" and "reviews" respectively.

"Star ratings" is a rating system which measures the degrees of satisfaction relates to the number of stars (from the scale 1 to 5), by using the ordinal levels of measurement of the qualitative data. The more stars are given indicates a higher degree of satisfaction. For example, one star (i.e. the scale of 1) represents an extremely low rated and low satisfaction of the customer.

"Reviews" is a system where customers can add further text-based messages, express their further comments, opinions and informations about the product. On the one hand, the "reviews" system tends to describe the product more specifically and provide with detailed information by using textual content compared with the "star ratings" system, on the other hand however, it remains a problem for us to convert the textual evaluation into quantitative data in the convenience of further data processing and numerical analysis. By analyzing these two sorts of data, companies can gain useful insights of the current and future marketplace, hence can make forecasts and inferential business decisions.

1.2 Problem analysis

The Sunshine Company is now planning to launch three new products online: a microwave oven, a baby pacifier, and a hair dryer. As the consultants of the Sunshine Company, our team focus on the provided data of the transaction history of Amazon from 2003 to 2015, related to these three products. Our aim is to identify and describe the relationships, key patterns, measures and parameters according to the given transaction history, especially based on the combination of the datas and time pattern.

2 Assumptions

In order to concretize and simplify the model, we make the following assumptions:

Assumption 1
For the microwave ovens, hair dryers and the baby pacifiers, according to the given data: customer_id, we assumer there exists no repeat customer (i.e. no repeat purchase).

Assumption 2
A category of a product with a large scale of sales volume indicates that, it is impossible to have relatively low mean of star ratings. The reasons are that, almost everyone prefer product with good reputation, if the rating of a product is low, there wouldn't be such many people to purchase it. Besides, the large scale of sales volume can offset the effect of outliers.

Assumption 3
The data of sales volume at present has already reflected the effect of "star ratings" and "reviews" in the past intrinsically and completely.

3 Mathematical model of sentiment analysis

3.1 Data pre-processing

The "star ratings" and "reviews" are both contained in three products respectively, thus we can obtain $3\times2=6$ groups of data. We firstly process the data groups of "reviews", the steps are shown as following:

1. We set up three string vectors which respectively represent the "reviews" columns of the 3 products.

2. Some of the "reviews" have no actual meaning for the program, such as emoticons or texts written by minority languages, hence we remove those elements in the columns of "reviews", and the corresponding elements in the columns of "star ratings", due to "reviews" and "star ratings" form a pair of **Variables**.

3. By using the sentiment analysis (details will be shown as 2.2 below) in **R Language**, we can therefore convert the text informations (reviews) into numerical data measures, in other words, the string vectors have been converted into numerical vectors, and all the elements in the numerical vectors are rated within the interval of $[-1, 1]$.

4. Some NaNs (Not a Number that cannot be identified by the program) are still contained in the processed numerical vectors, we should further remove them and the corresponding "star ratings" as above.

By doing the steps above can we obtain the so-called "effective vectors".

Then we process the data groups of "star ratings". We can directly categorize and aggregate the data of "star ratings" due to the data is originally numerical, then we plot the following bar chart:

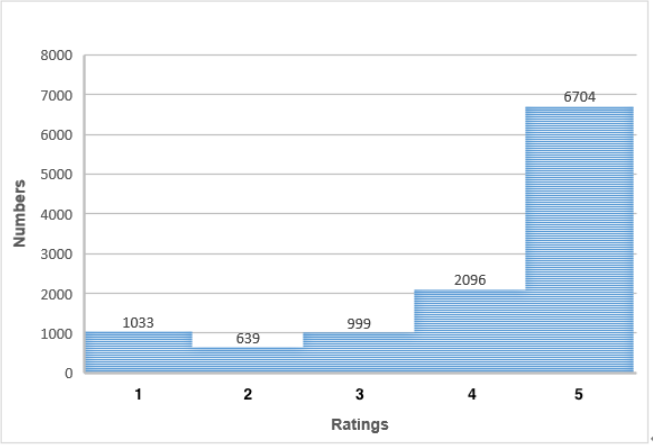

Figure 1: Aggregated amount of votes for each scale

3.2 Sentiment analysis

In the sentiment analysis, The *bing* lexicon divides texts into **positive** and **negative** sentiments. For each comment or "review" (that is the element in the string vector) can the program produce one or several **"sentiments"**, and these sentiments can be rated comprehensively to a score by the rating algorithm [3].

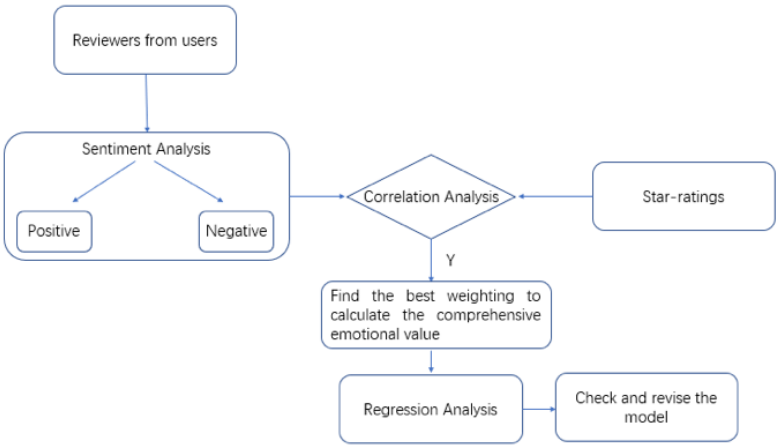

Figure 2: Internal logics and processes of sentiment analysis

Rating and reviews are direct sources which can reflect the degree of satisfaction of the customers. In order to set up a data measure to help the company to identify and to estimate, we consider using a combination of rating and the scores of comments as variables. Sentiment analysis is needed in this model to transform the content of the reviews into scores within a certain range between $[-1, 1]$.[7]

Firstly, the ratings from 0-5 stars are already given in the dataset, and another value is derived from the users comments through SentimentAnalysis package in R 3.6.3.

We use one sample two-sided test at the emotional value we extract from the reviewers and the null hypothesis is $H_0 : \mu = 0$ against alternative hypothesis $H_1 : \mu \neq 0$. P-value of the t-test is smaller than $2.2 \times e^{-16}$ and the 95% confidence interval is $[0.2315979, 0.2403969]$. We could safely draw the conclusion that the true mean should be bigger than 0.

The test here is called **Kolmogorov-Smirnov normal test**, which is used to compare the frequency distribution function f(x) and the theoretical distribution function g(x), or the distributions of two observation value. The null hypothesis H_0: "the observation data is distributed theoretically" or "the above two data are distributed uniformly".

$$\boxed{\textit{The actual observation value}: \ D = max\,|f(x) - g(x)|}$$

If the actual observation value: D>D(n,α), then reject H_0; otherwise accept H_0.

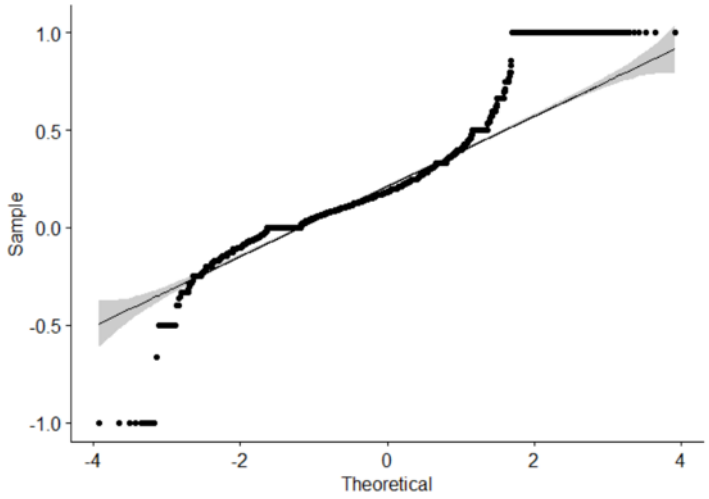

Figure 3: Kolmogorov-Smirnov normal test to check the distribution of the whole sample

Similarly, we apply the same test to the given "star ratings" and the 95% confidence level is $[4.092191, 4.139855]$. The mean for this sample is 4.116023, which is bigger than the medium star rating 3. To simplify, we assume that the samples are independent and identically distributed (i.i.d.).

Considering the fact inconsistency between comments and ratings might occur, consistency test is necessary before distributing the weight into the two factors. Correlation analysis was conducted between the comprehensive emotional value and the accurate rating.

A simple model is adopted here using **Pearson correlation coefficient**, which is the covariance of the two variables divided by the product of their standard deviation and as shown below:

$$\boxed{\rho_{x,y} = \frac{cov(x,y)}{\sigma_x \cdot \sigma_y}}$$

Here, cov(x,y) is the covariance and σ_x, σ_y are the standard deviation of x and y respectively. The absolute values of Pearson correlation coefficient are less than or equal to 1. Before actually calculating the correlation coefficient, we notice the fact that the variables have different scales. Therefore, feature scaling is applied to normalize the features of data.

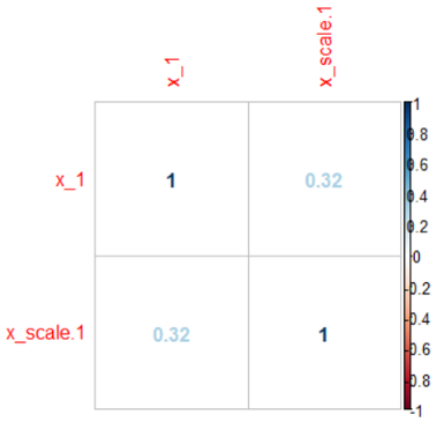

Figure 4: Correlation heatmap

3.3 Predicting the success or failing product: Logistic regression

3.3.1 Defining the success or failing product

We denote $\bar{x}$ and $\bar{y}$ as the overall means of "star ratings" values and "reviews" values of all the same products (e.g. all the microwave ovens) respectively. However, as for that same product can we categorize it by the **product_parent**, (e.g. product_parent: 732252283 is one category of hairdryers).

For each category of a product, we denote the order number by i, where i=1,2,...,m, and we denote x_i as the mean of "star ratings" values of the same category i of the product. For example, as for the one category of the hair dryers, let's say the k^{th} category ($1 \leq k \leq m$) whose **product_parent: 732252283**, has its unique means of "star ratings" value, denoted by x_k.

For a fixed category i of a product, we denote $y_{i,j}$ as a single "reviews" value from the same category of the product. **Moreover**, where j=1,2,...,n. Therefore, as for the "reviews" values, we have $y_{i,j}$ to denote **a single order in one category of a product**. From the previous example, product_parent: 732252283 is the k^{th} category of the hair dryers, and this category has 587 orders before data pre-processing, then for such single order in such category, it has $y_{i,j}$ for i=k, and j=1,2,...,587.

In order to define whether a categorized product is success or not, we denote J_i as a **comprehensive numerical measures** of the category i of one product, which are combined with the ratings-based measures (x_i) and the text-based measures ($y_{i,j}$) of the i^{th} category.

For a fixed category i of a product, we regard the subtraction of x_i and $\bar{x}$ as the "star ratings" **contribution** to its comprehensive numerical measure J_i, for the same reason, the subtraction of $y_{i,j}$ and $\bar{y}$ is the "reviews" contribution to J_i. Notice that the contributions can be positive or negative, due to negative feedback also provides negative impact to that product category.

The fixed category i contains n orders, hence we have to multiply n before the contribution of "star ratings", and we have to automatically sums the contributions of "reviews" from 1 to n. For the accuracy of the model, we also need to add coefficients α and β before both the contributions, where $\alpha, \beta \in (0, 1)$. Therefore we have:

$$J_i = \alpha n(x_i - \bar{x}) + \beta \sum_{j=1}^{n} (y_{i,j} - \bar{y}) \tag{1}$$

For the "reviews", we also detect the **helpful_votes** for some text comments. The amount of the helpful_votes for a comment indicates that, there are extra people at that amount who have also expressed the same nature of "review". Therefore, we need to multiply $(1+v_j)$ before the "reviews" contribution, where v_j denotes the amount of the helpful_votes for the j^{th} review, and 1 represents the reviewer himself. Although the helpful_votes show the same nature of feelings to the reviewer, the weights of the votes and the reviews themselves should never be equivalent, it should be another coefficient γ before the v_j, where $\gamma \in (0, 1)$. Therefore we have (2) from (1):

$$J_i = \alpha n(x_i - \bar{x}) + \beta \sum_{j=1}^{n} (1 + \gamma v_j)(y_{i,j} - \bar{y}) \tag{2}$$

We consider any category to be successful if the value of J is larger than 0, otherwise to be failed.

3.3.2 Variable re-statement

Here, we want to re-state our variables:

J_i: the comprehensive numerical measures of the category i of one product.

x_i: the mean of "star ratings" values of the same category of a product, where i=1,2,...,m, and m equals to the total amount of categories of a product.

$y_{i,j}$: a single "reviews" value from the same category i of the product, where j=1,2,...,n, and n equals to the total amount of reviews for category i.

$\bar{x}$: overall means of "star ratings" values of all the same products.

$\bar{y}$: overall means of "reviews" values of all the same products.

v_j: the amount of the helpful_votes for the j^{th} review.

α, β, γ: coefficients which $\in$ (0,1), we set α=$\frac{1}{3}$, β=$\frac{2}{3}$, and γ=$\frac{1}{10}$ as modified value.

Notice that, we remove some extreme small samples which contain one order, because small sample size may lead to **outliers** with high probabilities.
By doing the computation of the above formula (2), we get the histogram of the distribution of **J** and the histogram of the distribution of **review**.

3.3.3 Data processing and analysis

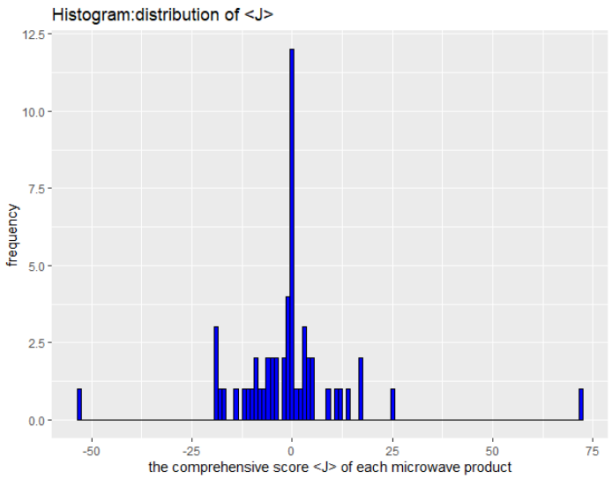

Figure 5: Histogram of the distribution of review

According to the above algorithm, we can generate a list of J: comprehensive numerical measure for each parent product. By arranging it in descending order in *Excel* can we obtain the five most successful product and their relevant information.

No.	id_array	count	avg_star	avg_review	J-score
1	423421857	391	3.864450128	0.06637031	72.1580544
2	827502283	76	4.118421053	0.00764865	24.5473954
3	295520151	74	3.554054054	-0.057685453	17.4843058
4	771401205	78	4.205128205	-0.056078407	16.8849981
5	523301568	74	4.5	-0.030643467	13.9180875

Figure 6: List of value of J

Based on the previous assumptions, the text-based and ratings-based measures, our aim is to fit a model which can best indicate a potentially successful or failing product. A new binary variable is created to indicate the state of this parent product. **"Success" is denoted by "1" and "Failure" is denoted by "0"**. The value of this variable is based on comprehensive numerical measure "J" we have already obtained. More specifically, is the comprehensive score for the i^{th} parent product. The distribution scatterplot of is shown as follows:

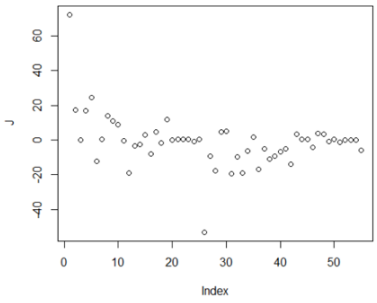

Figure 7: histogram of the distribution of J

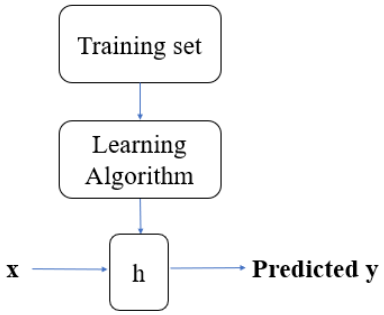

Figure 8: histogram of the distribution of review

We define J-score over 0 as "success" and below 0 as "failure" subjectively. Therefore we get the graph as below. In the following graph, the horizontal axis "diff-review" represents $y_{i,j} - \bar{y}$, and the vertical axis "diff-star" represents $x_i - \bar{x}$.

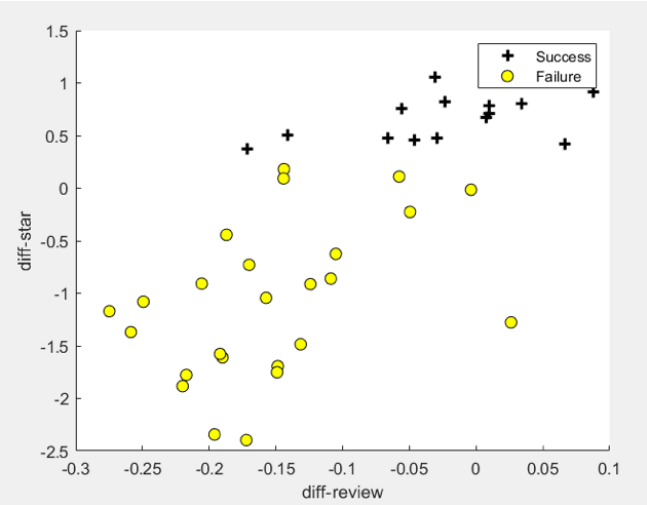

Figure 9: scatter plot of state variable

3.3.4 Fitting the model: Introducing the Logistic regression

Here in order to fit the model, the idea of supervised learning is introduced, which indicates mapping an input to an output based on example **input-output pairs** [4]. Here we already have a set of labeled training data. To establish notation for future use, we will use **X** to denote the "input" variables, also known as **input features**. In this case **X** is a matrix which contains column **diff-star** and **diff-review**, and y in this case is the state variable {0, 1}, also known as the target variable we are trying to fit and predict [5].

Before we build the model, we should introduce the **sigmoid function**:

$$g(z) = \frac{1}{1 + e^{-z}} \tag{3}$$

Notice that g(z) towards 1 as $z \longrightarrow \infty$, and g(z) tends towards 0 as $z \longrightarrow -\infty$.

Moreover, the logistic regression is defined as:

$$h_\theta(x) = h(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}} \tag{4}$$

The **cost function of logistic regression** is defined as follows:

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \left[-y^{(i)} log(h_\theta(x^{(i)})) - (1 - y^{(i)}) log(1 - h_\theta(x^{(i)})) \right] \tag{5}$$

As an illustration, here $\{x^{(i)}$, i=0,1,2$\}$ denotes the information for each product and we are keeping the convention of letting x_0=1 by adding a column of element 1. The gain and momentum update expression for a **gradient descent** are calculated by differentiating the cost J(θ) [5].

We could obtain θ by the following iterative algorithm to minimize J(θ) [6]:

$$Repeat : \{ \quad \frac{\partial J(\theta)}{\partial \theta_j} = \frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)} \tag{6}$$

$$\theta_j := \theta_j - \frac{\alpha}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)} \quad \} \tag{7}$$

Here α is the **learning rate of the gradient descent algorithm** and m is the **number of the samples**. By doing this, we can first calculate the initial gradient and after iteration we could obtain θ under this circumstance. ($\vec{\theta} = [-9.1195, 4.1623, 35.2655]$)

Using the parameter obtained we could get the prediction formula after vectorization:

$$y = X\theta = \theta_0 + \theta_1 x_1 + \theta_2 x_2 \tag{8}$$

With this formula, we can plot the **decision boundary** and check the accuracy of the fitting model as following:

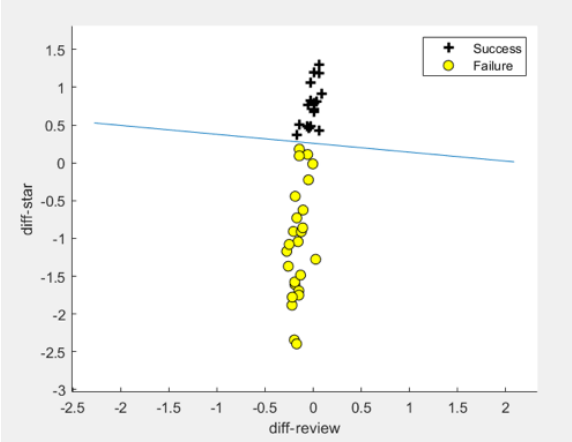

Figure 10: Decision boundary

3.4 Influence of specific words

In our model of sentiment analysis, whether the word is positive of negative is associated with the review rating. Positive words are connected to positive ratings, while negative words are connected to negative ratings. Meanwhile, star rating levels are associated with review rating positively. Therefore, specific quality descriptors of text-based reviews are strongly associated with rating levels.

4 Influence of specific star ratings

We want to discover the further influence of specific star ratings. Here, we take the situation of high ratings as an example. We aim to find whether a series of concentrated five stars will influence the subsequent star ratings.

Firstly, if there are at least four five-star ratings in consecutive five ratings, we define it as a series of high ratings. Then, we need to check whether it would influence the next star ratings. An array of the next star ratings need to be constructed. We should classify all star ratings of the product. For an array of $(r_i, r_{i+1}, r_{i+2}, r_{i+3}, r_{i+4})$ in succession, if it exists ≥ 4 five-star ratings, then r_{i+5} will be added to the array (i=1,2,...,p-5; where p equals to the amount of the product). After the construction of the new array, we can calculate the mean value of the new array and compare it with the mean value of all star ratings. All the values we derive from the sample are listed below as a flowchart.

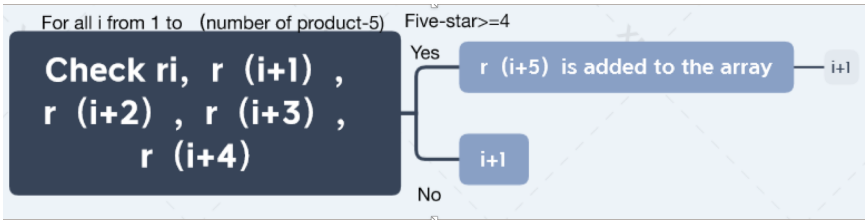

Figure 11: The flowchart to process the sample

Mean of star rating of microwave in the sample: 3.443548
Mean of star rating of microwave in new array: 3.467769
Mean of star rating of hair dryer in the sample: 4.116023
Mean of star rating of hair dryer in new array: 4.126028
Mean of star rating of pacifier in the sample: 4.304341
Mean of star rating of pacifier in new array: 4.319911

Now, we take x as the mean of star rating of the product in the sample, $\tilde{x}$ as the mean of star ratings of the product in the new array. Then, define

$C = \frac{\tilde{x}-x}{x}$ as the relative change which can illustrate how it change from the original mean. We calculate the relative change of these three product.

microwave: 7.03%
hair dryer: 2.43%
pacifier: 3.62%

According to the statistics above, ratings of the three products have all increased slightly, hence, a series of concentrated high ratings will influence the next ratings **positively**.

5 Time-based measures to detect changes in reputation

5.1 Time series decomposition and an overview of components of decomposition

From an overview of the results after processing the dataset by months, we found that the information before year 2011 is insufficient. **So the average stars and emotional values of reviews per month are taken after year 2011**. Given data from 2011 to 2014, our aim is to fit the model and give an appropriate estimation of the reputation of year 2015.

The average scores of stars and reviews of year 2015 can be used to make comparison. An overall increasing trend can be seen in the numeric values of the average month score of stars and reviews. In order to visualize this, we decompose a time series into its constituent components, which are usually a trend component and an irregular component, and if possible, a seasonal component [1].

$$x_t = m_t + S_t + R_t \tag{9}$$

where x_t is the time series at time t, m_t is the trend, S_t is the seasonal effect, and R_t is the remainder.

However, in most cases, a multiplicative decomposition model is better based on emprical experience.

$$x_t = m_t \cdot S_t \cdot R_t \tag{10}$$

$$log(x_t) = log(m_t \cdot S_t \cdot R_t) = log(m_t) + log(S_t) + log(R_t) \tag{11}$$

By applying this decomposition model, we can see a general increasing trend in terms of average stars and average review scores per month, as shown below.

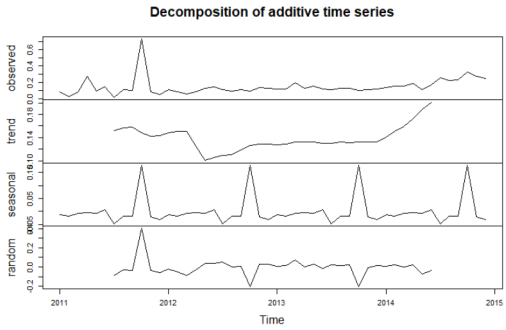

Figure 12: Decomposition of additive time series of star ratings

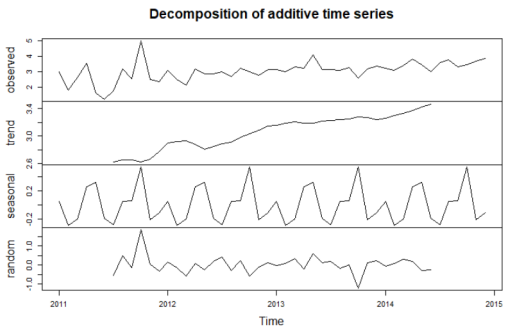

Figure 13: Decomposition of additive time series of reviews

5.2 Testing stationarity

5.2.1 Autocorrelation

Serial correlation is an important characteristic of time series. The autocorrelation between two variables is defined as:

$$Cor(X_{t+k}, X_t) = \frac{Cov(X_{t+k}, X_t)}{\sqrt{Var(X_{t+k})Var(X_t)}} \tag{12}$$

It is a dimensionless measure for the linear association between this two random variables. Since for stationary series, we require the moments to be non-changing over time. We can write the autocorrelation as a function of the lag k:

$$\rho(k) = Cor(X_{t+k}, X_t) \tag{13}$$

The aims in the forthcoming sections are to estimate these autocorrelations from observed times series data, and to study the estimates? properties.

5.2.2 Plug-In Estimation

Autocorrelation estimation is commonly done using plug-in approach. The formula is as follows:

$$\hat{\rho}(k) = \frac{\hat{\gamma}(k)}{\hat{\gamma}(0)}, \ for \ k = 1, ..., n-1 \tag{14}$$

$$where \ \hat{\gamma}(k) = \frac{1}{n}\sum_{s=1}^{n-k}(x_{s+k} - \bar{x})(x_s - \bar{x}), \tag{15}$$

$$with \ \bar{x} = \frac{1}{n}\sum_{t=1}^{n} x_t \tag{16}$$

Notice: n is the denominator which is independent of the lag. As a result, $\hat{\gamma}(0)$ is not an unbiased estimator to $\gamma(0) = \sigma_X^2$, however, as we stated above, it have sufficient reasons to do that. The estimate of the k^{th} autocorrelation parameter while plugging in the stated terms is:

$$\hat{\rho}(k) = \frac{\sum_{s=1}^{n-k}(x_{s+k} - \bar{x})(x_s - \bar{x})}{\sum_{t=1}^{n}(x_t - \bar{x})^2}, \ for \ k = 1, ..., n-1. \tag{17}$$

For stationary series, with the increase of k, autocorrelation will decrease and tend to zero. Therefore, to test stationarity, we can calculate autocorrelations for different k.

5.2.3 Autocorrelation of the monthly average star rating values

According to our analysis, we figure that the star rating values and the reviews values are both decreasing sharply and converge to 0, with the increasing of the value k. (We take the microwave oven as an example as shown below.) **Therefore, we make sure that this time series is a stationary time series.**

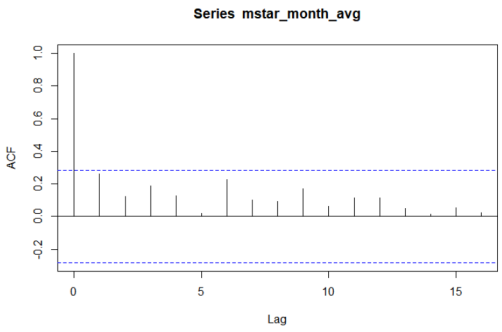

Figure 14: Correlogram of the average star ratings of the microwave oven

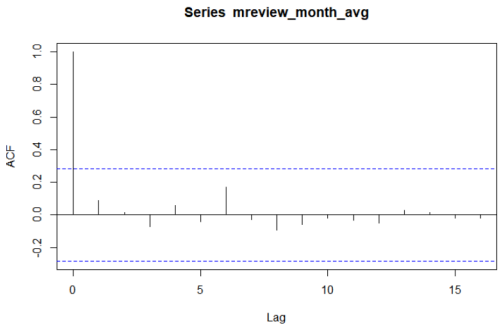

Figure 15: Correlogram of the average reviews of the microwave oven

5.3 Autoregressive Models

5.3.1 Definition and properties

An autoregressive model is a linear regression approach on the past instances, which is the most popular way of describing time series.
An autoregressive model of order p, abbreviated as AR(p), is based on a linear combination of past observations according to the following equation:

$$X_t = \alpha_1 X_{t-1} + \alpha_2 X_{t-2} + ... + \alpha_p X_{t-p} + E_t \tag{18}$$

Here, E_t comes from a White Noise process.
But firstly, the model must satisfy: AR(p) models must be only applied to stationary time series. Any possible trend or seasonal effect need to be removed first [8]. We must also ensure that the AR(p) processes are stationary.

5.3.2 Fitting

5.3.2.1 Model identification

Firstly, we should verify that the data show properties which make it reliable that they were generated from an AR(p) process. Particularly, the time series needs to be stationary, showing an ACF with approximately exponentially decaying envelope and a PACF with a recognizable cut-off at some lag p smaller than about 5-10. In the three properties, if anyone of them is extremely violated, it means that AR(p) may not have a satisfactory fitting, and there may be models which are more suitable [2].

5.3.2.2 Parameter estimation

Observed time series seem rarely centered and concentrated, therefore, it is incorrect to fit a pure AR(p) process. In reality, we can fit autoregressive models by assuming the shifted process: $Y_t = m + X_t$. Therefore, we have a regression-type equation with observations:

$$(Y_t - m) = \alpha_1(Y_{t-1} - m) + ... + \alpha_p(Y_{t-p} - m) + E_t \ for \ t = p+1, ..., n \tag{19}$$

Our target is to estimate the parameters: m, α_1,...,α_p, which are shown as following.

5.3.3 Yule-Walker Equations

An option for estimating AR(p) models is to plugging-in the sample ACF into the Yule-Walker equations. From the p×p linear equation system:

$$\rho(k) = \alpha_1\rho(k-1) + ... + \alpha_p\rho(k-p), \qquad for \ k = 1, ..., p \tag{20}$$

Hence, we can explicitly determine $\hat{\rho}(0), ..., \hat{\rho}(k)$ and then derive $\alpha_1, ..., \alpha_p$ from the linear equation system.

5.3.4 Apply Ar(p) models to the microwave dataset

After applying Yule-Walker Equation to the microwave dataset with order 3, we find the three coefficients (α_1, α_2,α_3) of star rating and review rating are (0.388, -0.0942, 0.2011) and (0.2225, 0.056, -0.1487). Then, we can use the model to predict the estimated star ratings and review ratings in 1.2015.

If the estimated value is higher than the true value, the reputation is decreasing. If the estimated value is lower than the true value, the reputation is increasing.

Star rating in 1.2015(estimate)=3.394081
Star rating in 1.2015(true)=3.763158
Review rating in 1.2015(estimate)=0.151014
Review rating in 1.2015(true)=0.245648

Both of the true ratings are higher than the estimation, so the reputation is increasing.

6 Strengths and weaknesses

6.1 Strength

1. We find a method to convert reviews into emotional scores, which is relatively accurate compared with some other methods.

2. Based on the comprehensive measure, we successfully arrange the information of the three products by descending order.
Afterwards, compared with the raw data, we can find the useful information contained in the product title of these commodities. Particularly, the comprehensive measure J is distributed around 0 so we can clear see the contribution of a certain parent product.

3. Logistics regression is used to fit the model here and to classify whether a parent product is successful or not. We use a quite standard procedure of logistic regression to realize classification, applying gradient descent.

4. With the use of autocorrelation, we are able to test the stationarity of the time series so that we can find a suitable model for the time series.

5. Time series for star rating and review rating is generally stationary and

the autoregressive model fits most stationary time series.

6. Yule-Walker Equation is associated with autocorrelation. Hence, it is easy to calculate the coefficients for the autoregressive model so that we can get a satisfactory fit with ease.

6.2 Weakness

1. In the models which requires text-based measures, we use Sentiment Analysis package in **R**.
However, we fail to convert emoticons which contain biased emotion to numeric values. During the data pre-processing, we remove these rows of data directly. Generally, this will not pose a threat to the final results because it only represents a relatively insignificant component of the samples.

2. Actually the star-ratings and review contents are highly associated with each other, which means high-ratings tend to appeal to more positive comments. However, in the model we put forward, we give same weighting ratios to all emotion values between [-1,1]. It is more reasonable to give distinguished weights to different levels of emotion values.

3. In 3.3.1 and 3.3.2 we take the value of coefficients $\alpha = \frac{1}{3}$, $\beta = \frac{2}{3}$ as example, we didn't try other sets of values during the calculation. However, the weighting ratios are flexible and edible, we can acquire more sets of coefficients to obtain different sets of J so that we can figure out which matches the real case best.

4. In the dataset for microwave, there are only 55 distinct parent products and empirically this is mot adequate to generalize a regression formula.

7 Sensitivity analysis

1. In a generalized situation, overfitting or underfitting might occur. Ridge and Lasso regularizations are methods which can reduce or shrink the coefficients in the ultimate regression so that the prediction changes less than it would have without the regularization [5].

This could be achieved using a regularized cost function in logistic regression:

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}\Big[-y^{(i)}log(h_\theta(x^{(i)})) - (1-y^{(i)})log(1-h_\theta(x^{(i)}))\Big] + \frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2 \tag{21}$$

2. To get a more accurate fitting result, we can use other algorithm such as **Burg's algorithm**. Burg's algorithm is based on the fact that any AR(p) process is a an AR(p) process if time is run in reverse order. Under this property, we can derive the coefficients by minimizing the forward and backward one step squared prediction errors:

$$\sum_{t=p+1}^{n} \left\{ (X_t - \sum_{k=1}^{p} \alpha_k X_{t-k})^2 + (X_{t-p} - \sum_{k=1}^{p} \alpha_k X_{t-p+k})^2 \right\} \tag{22}$$

3. We can find a method to consider reputation comprehensively which combines star rating and review rating together.

4. Residual analysis can be done to check the accuracy of our fitting.

8 Conclusion

This paper manages to find out quantitative and qualitative patterns hidden behind the dataset of the three products. Based on reviews, star-ratings, helpful votes and relevant variables, we build a criterion by which we can formulate the emotion value and further determine the product is good or not.

1. Comprehensive measure J is created based on reviews, stars and helpful votes after grouping each dataset by specific product_parent.

2. Using logistic regression to determine a certain product is successful or failing.

3. Taking time into consideration, using data from year 2011-2014 we group star-ratings and reviews by month and generate a time series. By decomposition, we could decompose the series into three elements and we could see a clear increasing trend in terms of stars and reviews per month.

4. In time series model, by testing the stationarity of the time series, we choose the autoregressive model which is suitable for stationary time series. After fitting the time series to autoregressive model, we can get the formula of the model so that we can use it to predict the rating in the future. By comparing the estimation and the real value, we can know whether the reputation is increasing or decreasing.

9 A letter to the marketing director of Sunshine Company

Dear Marketing Director of Sunshine Company:

We are writing to you regarding the most recommended products of microwaves, hairdryers and pacifiers. Our team has proposed text-based and rating-based measures to identity the quantitative and qualitative patterns of the dataset. A mathematical model is built to help determine whether a certain parent product is successful or failing in the market.

More specifically, we successfully transform the content of the reviews into emotional values which can indicate the degree of satisfaction of customers. We set up the definition of comprehensive numeric measure based on a few variables given. In the next step, in each dataset, we group the commodities by product_parent ID and arrange the data in descending order of comprehensive score(J score). A short list of the results are given as follows where you can get an overview of the Top 5 Seller for each product. Generally, "black" microwave with 0.5 cu.ft. is more popular among consumers. A more detailed file as following is attached in the appendix so that you can have a more explicit idea of what types or properties to choose.

id_array(product_parent)	product title	J-score
423421857	danby 0.7 cu.ft. countertop microwave	72.15805435
827502283	microwave cavity paint 98qbp0302	24.5473954
295520151	whirlpool stainless look countertop microwave, 0.5 cu. feet, wmc20005yd	17.48430577
771401205	whirlpool wmc20005yw countertop microwave, 0.5 cu. ft., white	16.8849981
523301568	ge microwave oven magnetron and diode kit om75p (10) part # wb27x10017	13.91808753

Figure 16: Microwave Top 5 Seller

id_array	J
235105995	27.13272514
593915883	20.03571958
614083399	12.81665485
833461643	12.10360683
153523919	8.819725282

Figure 17: Hairdryer Top 5 Seller

id_array	J
572944212	88.69511937
911821018	57.3870447
392768822	52.83056324
449026476	39.40721476
218693378	29.26182188

Figure 18: Pacifier Top 5 Seller

In the next step, we use logistic regression model to decide whether a product is successful or not. This model shows a pretty good result of prediction in the long term based on previous information.

Moreover, in order to decide whether the reputation is increasing or decreasing.

A new model is applied based on time series analysis. A simple overview of the results of data processing indicates that the numbers of comments experience a significant increasing trend in the most recent years in the given dataset, possibly owing to the prevalence of online shopping commodities. Further more, a combination of decomposition model and autoregressive model are used to give a more accurate estimation of the change of reputation. After comparing the estimated values and real values, we can detect whether the reputation is increasing or decreasing.

The recommended output from the model gives an acceptable measure based on ratings and reviews of the products. Multiple models are combined and compared to give you this recommendation. We are confident that this model is of applicable use in the future.

Sincerely,

MCM Team 2018546

10 Appendix A: Data for the J score of the example of microwave oven

	id_array	count	avg_star	avg_review	J
1	423421857	391	3.864450128	0.06637031	72.15805435
5	523301568	74	4.5	-0.030643467	24.5473954
2	827502283	76	4.118421053	0.00764865	17.48430577
4	771401205	78	4.205128205	-0.056078407	16.8849981
8	930071734	32	4.625	0.061684547	13.91808753
19	305608994	45	4.266666667	-0.023218407	11.65022198
9	109226352	79	3.924050633	-0.0291874	11.11602276
10	984005611	19	4.736842105	0.059483684	8.944320212
30	801135043	14	4.357142857	0.088086701	5.085583405
17	731025324	19	4.157894737	0.00974836	4.647672775
29	309267414	11	4.636363636	0.008243867	4.434110939
47	690479711	30	3.9	-0.045903174	3.64645264
48	721617315	13	4.230769231	0.009952798	3.497547902
43	572011672	12	4.25	0.033888002	3.496910468
15	459626087	41	3.951219512	-0.141533394	3.069592618
35	809249591	13	3.923076923	-0.066281773	1.503514957
44	632928046	1	5	0.136938775	0.610109721
7	692404913	44	3.818181818	-0.171577333	0.46168855
25	454581724	1	4	0.403605442	0.454554166
45	664466484	1	5	-0.12885213	0.432915784
23	149559260	1	5	-0.19567347	0.388368224
22	313983847	1	4	0.303605442	0.387887499
50	862802057	1	5	-0.196394558	0.387887499
21	665261008	1	4	0.02682653	0.203368225
53	788261054	1	4	0.013165335	0.194260761
52	542731946	1	4	-0.01067347	0.178368225
20	168181302	14	3.428571429	-0.003866505	-0.105979855
3	295520151	74	3.554054054	-0.057685453	-0.120009247
54	539049610	1	3	-0.046803244	-0.179051625
11	991090482	16	3.625	-0.144047516	-0.5687649
49	494028413	1	1	-0.108159264	-0.886622305
24	311592014	1	1	-0.146979593	-0.912502524
51	550562680	1	1	-0.645761905	-1.245024066
18	464779766	28	3.535714286	-0.144296506	-1.833319725
14	542519500	23	3.217391304	-0.049505163	-2.492950137
13	166483932	13	3	-0.187028379	-3.542955628
46	760984384	12	2.583333333	-0.108920716	-4.31222594
41	994339247	12	2.166666667	0.026115389	-4.898603767
37	215953885	14	2.714285714	-0.170078453	-4.990624701
55	379992322	11	2.363636364	-0.249332166	-5.788113305
34	155528792	11	2.272727273	-0.274772419	-6.308008493
40	981162112	15	2.4	-0.157393758	-6.79167952
16	838179571	12	1.833333333	-0.190024179	-7.96105365
27	147401377	10	1.1	-0.196032865	-9.118713727
39	486381187	33	2.818181818	-0.105156734	-9.192480406
32	784164614	15	1.866666667	-0.191867061	-9.80307921
38	522487135	15	1.666666667	-0.217151102	-11.05591963
6	565072108	28	2.535714286	-0.205594788	-12.31088766
42	618770050	24	1.958333333	-0.131508391	-13.98585469
36	242727854	27	2.074074074	-0.258686959	-16.98163408
28	494668275	26	1.692307692	-0.149141307	-17.76253534
12	943347999	49	2.530612245	-0.124164337	-18.96732534
33	921964554	21	1.047619048	-0.172167093	-19.18184468
31	392967251	25	1.56	-0.219877157	-19.36085584
26	544821753	80	1.75	-0.148640505	-53.0887839

Figure 19: Microwave(J score)

	id_array	count	avg_star	avg_review	J
6	235105995	407	4.257985258	0.029016662	27.13272514
15	593915883	159	4.283018868	0.105518321	20.03571958
24	614083399	146	4.253424658	0.062977164	12.81665485
114	833461643	83	4.506024096	0.023739367	12.10360683
99	153523919	44	4.659090909	0.029138531	8.819725282
91	286798751	29	4.655172414	0.172831373	8.553183592
49	290876515	78	4.269230769	0.085598229	8.434508228
50	328811288	253	4.201581028	0.005656689	8.16948316
40	392681682	74	4.324324324	0.045212309	7.368571629
70	74735317	94	4.244680851	0.052569174	7.325612201
60	888313825	59	4.169491525	0.157490301	7.246164905
7	195677102	140	4.164285714	0.032831784	5.316556814
8	582752797	30	4.4	0.120885089	5.257471137

Figure 20: Hairdryer(J score which less than 5)

11 Appendix B: Code

11.1 B_1: Calculating the comprehensive measure of microwave oven(By using R)

```
head(mreview.new)
total<-length(mreview.new$review)
total
id_array<-unique(mreview.new$product_parent)
head(id_array)
id_array<-as.vector(id_array)
id_array[1]

id_number<-length(id_array)
id_number
review_mean<-mean(mreview.new$review)
review_mean
star_mean<-mean(mreview.new$star)
star_mean

id<-as.vector(mreview.new$product_parent)
id[1]

count<-array(0,dim=c(55))
sum_review<-array(0,dim=c(55))
sum_star<-array(0,dim=c(55))
sum_votes<-array(0,dim=c(55))

J<-array(0,dim=c(55))
alpha<-1/3
beta<-2/3
gamma<-1/10
#Get the values for each product_id

for (j in 1:id_number){

  for (i in 1:total) {
    if (identical(id[i],id_array[j]))       #id_array has distinct elements with
      {count[j]<-count[j]+1          #count[] is the number of each product
      sum_review[j]<-sum_review[j]+ (mreview.new[i,5]*gamma+1)*(mreview.new[i,2]-
      sum_star[j]<-sum_star[j]+mreview.new[i,3]
     sum_votes[j]<-sum_votes[j]+mreview.new[i,5]
    }
  }
```

```
  J[j]<-alpha*(sum_star[j]-count[j]*(star_mean))+beta*sum_review[j]

}
mreview_new[50,2]

count
sum(count)
boxplot(J)
count[1]
#product
avg_star<-sum_star/count
avg_review<-sum_review/count
product_info<-data.frame(id_array,count,avg_star,avg_review,J)
head(product_info)
summary(product_info$count)

# count and J, count and rating
product_info %>% ggplot(aes(count))+
  geom_histogram(binwidth = 1,fill="blue",col="black")+
  xlab("the amount of each microwave product")+
  ylab("frequency")+
  ggtitle("Histogram:distribution of <count>")
product_info %>% ggplot(aes(J))+
  geom_histogram(binwidth = 1,fill="blue",col="black")+
  xlab("the comprehensive score <J> of each microwave product")+
  ylab("frequency")+
  ggtitle("Histogram:distribution of <J>")
#*(count[j]+sum_votes[j]/count[j])*(sum_review[j]/count[j]-review_mean)


write.csv(product_info,"C://Users//sz//Desktop//MCM2020//product_info.csv")
m<-cor(product_info$count,product_info$J)
m
corrplot(m)
avg_star_revised<-avg_star-star_mean
avg_star_revised
product_info<-data.frame(product_info,avg_star_revised)
state<-array(0,dim=c(55))
for (k in 1:55){
  if ((J[k]>0)&&(count[k]>=10))  {state[k]<-1}
    else {state[k]=0}
}state
product_info<-data.frame(product_info,avg_star_revised,state)
write.csv(product_info,"C://Users//sz//Desktop//MCM2020//product_info_v2.csv")
```

11.2 B_2: Time series analysis(Take microwave as example, by using R)

```
microwave_NEW<-microwave_NEW[-1,]
microwave_ts<-data.frame(microwave_NEW,mreview$review)
head(microwave_ts[,8])
microwave_ts<-data.frame(microwave_ts,microwave_ts[,8])
install.packages("lubridate")
library(lubridate)
Date<-strsplit(microwave_ts[,15],split="/")
head(Date)
year<-do.call(rbind, Date)[,3]
head(year)
month<-do.call(rbind, Date)[,1]
head(month)

microwave_ts<-data.frame(microwave_ts,month,year)

year_array<-unique(microwave_ts$year)
#
year_array_test<-c("2011","2012","2013","2014")
month_array_test<-c("1","2","3","4","5","6","7","8","9","10","11","12")
year_array_test[1]
star_month<-matrix(0,nrow=4,ncol=12)
review_month<-matrix(0,nrow=4,ncol=12)
count_month<-matrix(0,nrow=4,ncol=12)
review_month_avg<-matrix(0,nrow=4,ncol=12)
star_month_avg<-matrix(0,nrow=4,ncol=12)

write.csv(microwave_ts,"C://Users//sz//Desktop//MCM2020//microwave_ts.csv")
for (i in 1:length(year_array_test)) {
  for (j in 1:12){
    for (k in 1:1612){
      if ((year[k]==year_array_test[i])&&(month[k]==month_array_test[j])){


        count_month[i,j]<-count_month[i,j]+1
        review_month[i,j]<-review_month[i,j]+microwave_ts$mreview.review[k]
        star_month[i,j]<-star_month[i,j]+as.numeric(microwave_ts[k,8])
        review_month_avg[i,j]<-review_month[i,j]/count_month[i,j]
        star_month_avg[i,j]<-star_month[i,j]/count_month[i,j]



      }
```

```
    }
  }
}
count_month
star_month
review_month
star_month_avg
review_month_avg

mcont_month<-as.vector(t(count_month))
mcount_month
mstar_month_avg<-as.vector(t(star_month_avg))
mstar_month_avg
mreview_month_avg<-as.vector(t(review_month_avg))
mreview_month_avg
#Stationsay Test
#mstar_month_avg
#mreview_month_avg
acf(mstar_month_avg,plot=TRUE)
acf(mreview_month_avg,plot=TRUE)
startimeseries<-ts(mstar_month_avg,frequency=12,start=c(2011,1))
reviewtimeseries<-ts(mreview_month_avg,frequency=12,start=c(2011,1))
counttimeseries<-ts(mcount_month,frequency = 12,start=c(2011,1))
startimeseries
plot.ts(startimeseries)
plot.ts(reviewtimeseries)
plot.ts(counttimeseries)
#plot.ts(startimeseries)
#plot.ts(reviewtimeseries)
#plot.ts(counttimeseries)
startimeseriescomponents<-decompose(startimeseries)
startimeseriescomponents$seasonal
plot(startimeseriescomponents)
reviewtimeseriescomponents<-decompose(reviewtimeseries)
plot(reviewtimeseriescomponents)
counttimeseriescomponents<-decompose(counttimeseries)
plot(counttimeseriescomponents)
microwave_month<-data.frame(mcount_month,mreview_month_avg,mstar_month_avg)
microwave_month
write.csv(microwave_month,"C://Users//sz//Desktop//MCM2020//microwave_month.csv"
plot(log(startimeseriescomponents))
plot(log(reviewtimeseriescomponents))
#2015
star_month_check<-matrix(0,nrow=1,ncol=8)
review_month_check<-matrix(0,nrow=1,ncol=8)
```

```
count_month_check<-matrix(0,nrow=1,ncol=8)
review_month_check_avg<-matrix(0,nrow=1,ncol=8)
star_month_check_avg<-matrix(0,nrow=1,ncol=8)

for (j in 1:8){
    for (k in 1:1612){
      if ((year[k]=="2015")&&(month[k]==month_array_test[j])){

        count_month_check[1,j]<-count_month_check[1,j]+1
        review_month_check[1,j]<-review_month_check[1,j]+microwave_ts$mreview.re
        star_month_check[1,j]<-star_month_check[1,j]+as.numeric(microwave_ts[k,8
        review_month_check_avg[1,j]<-review_month_check[1,j]/count_month_check[1
        star_month_check_avg[1,j]<-star_month_check[1,j]/count_month_check[1,j]


      }

    }
  }
star_month_check
review_month_check
count_month_check
review_month_check_avg
star_month_check_avg

mcount_month_check<-as.vector(t(count_month_check))
mcount_month_check
mstar_month_check_avg<-as.vector(t(star_month_check_avg))
mstar_month_check_avg
mreview_month_check_avg<-as.vector(t(review_month_check_avg))
mreview_month_check_avg

startimeseries_check<-ts(mstar_month_check_avg,frequency=8,start=c(2015,1))
reviewtimeseries_check<-ts(mreview_month_check_avg,frequency=8,start=c(2015,1))
counttimeseries_check<-ts(mcount_month_check,frequency = 8,start=c(2015,1))
startimeseries_check
plot.ts(startimeseries_check)
plot.ts(reviewtimeseries_check)
plot.ts(counttimeseries_check)
microwave_month_2015<-data.frame(mcount_month_check,mreview_month_check_avg,msta
microwave_month_2015
write.csv(microwave_month_2015,"C://Users//sz//Desktop//MCM2020//microwave_month
```

11.3 B_3: MATLAB(Logistic regression)

```
%%
X=(productinfov3(:,[5,7]))
y=(productinfov3(:,9))
% Find Indices of Positive and Negative Examples
pos = find(y==1);
neg = find(y == 0);
% Plot Examples
%%

%plot(X(pos, 1), X(pos, 2), 'k+','LineWidth', 2, 'MarkerSize', 7);
%plot(X(neg, 1), X(neg, 2), 'ko', 'MarkerFaceColor', 'y','MarkerSize', 7);
%figure3 = figure('Color',[1 1 1]);
plotData(X, y);

% Labels and Legend
xlabel('diff-review')
ylabel('diff-star')

% Specified in plot order
legend('Success', 'Failure')
%%
sigmoid(0)
m=42
n=2
X = [ones(m, 1) X];
X
initial_theta = zeros(n + 1, 1);

initial_theta
%%
[cost, grad] = costFunction(initial_theta, X, y);
fprintf('Cost at initial theta (zeros): %f\n', cost);
%%
disp('Gradient at initial theta (zeros):'); disp(grad);
%%
options = optimoptions(@fminunc,'Algorithm','Quasi-Newton','GradObj', 'on', 'Ma

%  Run fminunc to obtain the optimal theta
%  This function will return theta and the cost
[theta, cost] = fminunc(@(t)(costFunction(t, X, y)), initial_theta, options);

fprintf('Cost at theta found by fminunc: %f\n', cost);
%%
disp('theta:');disp(theta);
theta
```

```
%%

% Plot Boundary
plotDecisionBoundary(theta, X, y);
% Add some labels
hold on;
% Labels and Legend
xlabel('diff-review')
ylabel('diff-star')
% Specified in plot order
legend('Success', 'Failure')
hold off;

%%
prob = sigmoid([1 0.8 0.6].* theta);

prob
%%
p = predict(theta, X);
fprintf('Train Accuracy: %f\n', mean(double(p == y)) * 100);
```

References

[1] Avril Coghlan. *A Little Book of R For Time Series*. 2 edition, September 2018.

[2] Marcel Dettling. *Applied Times Series Analysis*. Institute for Data Analysis and Process Design, May 2016.

[3] Xu xin Ma Songyue. Study on user online evaluation based on sentiment analysis of comments: Taking douban.co movie as an example. *Library and Information Service*, 60(10):95–102, October 2016.

[4] Ameet Talwalkar Mehryar Mohri, Afshin Rostamizadeh. *Foundations of Machine Learning*. Number 9780262018258. The MIT Press, 2012.

[5] Andrew Ng. Cs229 lecture notes, machine learning. Stanford University.

[6] Rozaida Ghazali Norhamreeza Abdul Hamid, Nazri Mohd. Nawi. The effect of adaptive gain and adaptive momentum in improving training time of gradient descent back propagation algorithm on classification problems. *International Journal on Advanced Science, Engineering and Information Technology*, 1(2):178–184, January 2011.

[7] Yang Sinan Zhang Hongli, Liu Jiying and Xu Jian. Predicting online users' ratings with comments. *Data Analysis and Knowledge Discovery*, (8):48–58, 2017.

[8] Yanchang Zhao. *R and Data Mining: Examples and Case Studies*. Number 0123969638. Academic Press, 1 edition, December 2012.