基于文本情感分析的商品评价模型

**Summary**

With the rapid development of the information age, the Internet and modern information technology are increasingly developed. The network has become the most convenient platform for data sharing, which also marks the arrival of a new era -- the era of big data. In the era of big data, everyone is the creator and user of network data. Through big data, we can easily describe everyone's behavior habits and reflect the laws of things. E-commerce is the product of big data era. Founded in 1995 in Seattle, Washington, Amazon is the largest online e-commerce company in the United States. In the online marketplace it created, Amazon provides customers with an opportunity to rate and review purchases. In order to use these data to gain insights into the markets in which they participate, the timing of that participation, and the potential success of product design feature choices，it’s very meaningful to build a model to analyze user comments and ratings.

firsly, 为了帮助阳光公司提供的三个产品在亚马逊商城的销售取得成功，本文利用机器学习中的LightGBM算法建立模型, 通过自然语言情感分析技术将评论文字映射到数值空间, 同时对预处理之后的数据进行特征工程, 以帮助模型更好的分析数据, 建立模型后通过模型评价得到模型的F1值均在0.9左右, 表明模型建立的非常成功, 同时我们得出了数据中各指标的重要程度, 最终发现特征review\_body(评论), rate(好评率), review\_date(评论日期)和total\_votes(总投票数)对一个产品的评级影响很大, 故公司应注重相应的数据信息来成功运营产品.

Firstly, in order to help the sales of three products provided by sunshine company succeed in Amazon mall, this paper uses lightGBM algorithm which is popular in machine learning to build a model, through natural language emotion analysis technology, the comment text is mapped to the numerical space. At the same time, the data after preprocessing is feature engineered to help the model analyze the data better After the model is established, the F1 value of the model is about 0.9 through model evaluation, which indicates that the model is very successful. At the same time, we get the importance of each index in the data. Finally, we find that the characteristics review\_body, rate, review\_date and total\_votes have a great impact on the rating of a product. Therefore, the company should pay attention to the corresponding data information to successfully operate the product.

secondly, 针对第二问的a小问, 在第一问时建立的模型就已经综合考虑了各种相关数据, 一旦阳光公司的产品开始销售, 根据反馈的评价信息, 将相关数据输入到模型中得到预测评分, 再根据预测评分和真实评分来分析数据得到相关结论. 对于b小问, 本文先通过熵权法确定了综合评级和评论的评价score结果来描述产品的声誉, 并通过月份销售量, 好评率, 差评率的统计数据绘制统计图进行基于时间模式的分析, 发现xxxxxxx; 由于前边两个模型都结合了相关的文本和评级的度量, 故c小问通过模糊综合评价方法利用问题一模型结论得出相关指标和b小问得出的score进行综合评判得到相应指标的权重, 利用权重计算综合得分, 评判一个产品的潜在成功与否; d小问要求分析特定的评级是否会引发某种类型的评论, 通过计算二者的斯皮尔曼相关系数和实际数据发现, 特定的评级会引发某种类型的评论; 最后, 通过构建特定描述符词典, 统计相关描述符的词频, 并利用热力图和词云可视化分析得出, “never”等否定词汇与一星级密切相关, 而”love”,”like”,”great”等积极词汇常出现在4星或5星评级中.

Secondly, for the A question of the second question, the first question’s model has comprehensively considered all kinds of relevant data. Once the products of sunshine company start to sell, input the relevant data into the model according to the feedback evaluation information to get the prediction score, and then analyze the data according to the prediction score and the real score to get the relevant conclusions. In this paper, entropy weight method is used to determine the evaluation score results of comprehensive rating and comment to describe the reputation of products, and through the statistical data of monthly sales volume, favorable rate and poor rating rate to draw a statistical chart for the analysis based on the time pattern, it is found that XXXXXX. The two models in the front combine the relevant text and rating measurement, Therefore, C the first mock exam uses fuzzy comprehensive evaluation method to draw the relevant index and the score obtained by B's small question by using the problem model conclusion, and gets the weight of the corresponding index, calculates the comprehensive score by weight, judges the potential success of a product; D asks whether the specific rating will cause some kind of comment. By calculating the Spearman correlation coefficient and the actual data, it is found that a specific rating will lead to a certain type of comment. Finally, by building a dictionary of specific descriptors, the word frequency of the relevant descriptors is counted, and the thermodynamic diagram and word cloud visualization analysis are used to draw the conclusion that "never" and other negative words are closely related to a star, while "love", "like", ”Positive words such as "great" often appear in 4 - or 5 - star ratings.

Finally, this paper summarizes the above data mining and analysis results into a report, which is provided to the marketing director of sunshine company by letter. I sincerely hope that this result can provide the best help for the company.

Keyword：

**Content**

1. **Introduction**

**1.1 Problem Formulation**

随着信息技术的持续大爆发，数据量也正在呈爆炸性增长的趋势，如何在大数据时代充分利用数据提取出有用的信息，这是一个非常具有意义的课题。亚马逊公司作为美国最大的电子商务公司，对用户的行为数据十分重视，该公司在其网络购物商城中为消费者提供了评分和评论的机会，本文将利用这些数据建立模型提取信息，为阳光公司新推出的三个产品上市及售后保驾护航。

With the continuous explosion of information technology, the amount of data is also growing explosively. How to make full use of the data to extract useful information in the era of big data is a very meaningful topic. As the largest e-commerce company in the United States, Amazon attaches great importance to the user's behavior data. The company provides consumers with the opportunity to score and comment in its online shopping mall. This paper will use these data to build a model to extract information, which will escort the listing and after-sales of three new products launched by sunshine company.

该题目设立的问题可以分为两个部分，其中第一部分仅包含一个问题，第二部分包涵6个问题，具体问题如下。

The question of this topic can be divided into two parts. The first part contains only one question, and the second part contains six questions. The specific questions are as follows.

第一部分:

Part I:

1. 分析吹风机、微波炉、奶嘴这三种产品的数据集，定量地测定评定星级、评论和Helpfulness Rating之间的数学关系，帮助阳光公司的三个新的上市产品获取成功。

a. The data sets of hair dryer, microwave and pacifier are analyzed, and the mathematical relationships among star rating, comment and helpfulness rating are measured quantitatively to help the three new products of sunshine company succeed.第二部分：

Part II:

1. 提供一种可以根据评级和评论数据提取信息的方法，当推出的三个产品在网络上销售后，阳光公司可以利用这种方法追踪产品销售情况。

a. Provide a method to extract information based on rating and comment data. When the three products launched are sold on the Internet, sunshine company can use this method to track product sales.

b.分别讨论着三个产品的数据基于时间的度量和模式，根据数据随时间的变化，分析商品声誉的变化。

b. This paper discusses the time-based measurement and model of three products' data respectively, and analyzes the change of commodity reputation according to the change of data over time.

c.确定一种基于文本的度量方法以及基于星级的度量方法组合，探寻产品是成功还是失败。

c. Determine a text-based measurement method and a combination of star based measurement methods to explore whether the product is a success or failure.

d.探寻客户是否具有从众心理，客户的情绪是否会随着他人的评价而被调动。

d. To explore whether customers have herd mentality and whether customers' emotions will be mobilized with the evaluation of others.

e.判断文本的特征词与评级水平是否密切相关

e. Judge whether the characteristic words of the text are closely related to the rating level

f.向阳光公司的市场总监写一封信推荐我们团队的测评结果并讲清楚理由。

f. Write a letter to the marketing director of sunshine company to recommend the evaluation results of our team and explain the reasons.

**1.2 Model Goals**

Sunshine公司的吹风机、微波炉和奶嘴这三种产品即将上市。本文建立模型期望达到的目标是，合理利用数据提取信息，帮助sunshine公司能够根据产品的评论和评级追踪产品的销售和声誉的变化情况，以及判断产品评级和评论对消费者的影响。总体来说，我们团队建模的目的是为sunshine公司制定出正确的销售策略做出贡献。

Sunshine's hair dryer, microwave and pacifier are coming soon. The goal of this model is to use data to extract information reasonably, to help sunshine company track the changes of sales and reputation of products according to product reviews and ratings, and to judge the impact of product ratings and reviews on consumers. In general, the purpose of our team modeling is to contribute to making the correct sales strategy for sunshine company.

1. **Assumptions**

为确保数学模型的建立顺利进行，现对模型做出以下假设：

In order to ensure the smooth establishment of the mathematical model, the following assumptions are made for the model:

1. 假设评论是真实的，不存在刷好评和故意抹黑。

1. It is assumed that the comments are true and there is no praise or intentional smear.

2.假设评论不存在反讽情况。

2. Suppose there is no irony in the comment.

3.假设评论使用的都是常用单词，如good, recommend等

3. Suppose that comments use common words, such as good, recommended, etc.

4.假设产品的质量保持不变

4. Assume the quality of the product remains unchanged.

5.假设客户对同一种类的产品的观点一致，如用户对hair\_drye下的产品的观点相同的。

5. It is assumed that customers have the same views on the same kind of products, for example, users have the same views on the products under hair dryer.

1. **Symbolic Explanation**
2. **Methodology**

**4.1** **GBDT algorithm**

GBD是一个集成多个决策树的机器学习模型. GBD is a machine learning model integrating multiple decision trees GBDT全称梯度下降树，在传统机器学习算法里面是对真实分布拟合的最好的几种算法之一,其具有有效性,准确性,可解释性, 在众多机器学习任务中取得了优越的结果. GBDT’s full name is gradient descent tree, and is one of the best algorithms to fit the real distribution in the traditional machine learning algorithm. It is effective, accurate and interpretable, and has achieved superior results in many machine learning tasksgbdt通过多轮迭代,每轮迭代产生一个弱分类器模型，每个分类器在上一轮分类器的残差基础上进行训练。对弱分类器的要求一般是足够简单，并且是低方差和高偏差的。Through multiple iterations, each iteration produces a weak classifier model, and each classifier is trained on the basis of the residual of the previous one. The requirements for weak classifiers are generally simple enough, with low variance and high deviation.因为训练的过程是通过降低偏差来不断提高最终分类器的精度。然而GBDT存在着致命的缺点—速度很慢, 需要对数据进行无数次的遍历.导致其应用有限. Because the training process is to reduce the deviation and to improve the accuracy of the final classifier. However, GBDT has a fatal disadvantage -- it is very slow and needs to traverse the data countless times, resulting in its limited application

**4.2 Fuzzy comprehensive evaluation model**

模糊综合评价方法是模糊数学中应用的比较广泛的一种方法。在对某一事务进行评价时常会遇到这样一类问题，由于评价事务是由多方面的因素所决定的，因而要对每一因素进行评价；在每一因素作出一个单独评语的基础上，如何考虑所有因素而作出一个综合评语，这就是一个综合评价问题。模糊综合评价是对受多种因素影响的事物做出全面评价的一种十分有效的多因素决策方法，其特点是评价结果不是绝对地肯定或否定，而是以一个模糊集合来表示。

Fuzzy comprehensive evaluation method is widely used in fuzzy mathematics. In the evaluation of a certain transaction, such a kind of problem is often encountered. Because the evaluation transaction is determined by many factors, it is necessary to evaluate each factor. On the basis of each factor making a separate comment, how to consider all factors and make a comprehensive comment is a comprehensive evaluation problem. Fuzzy comprehensive evaluation is a very effective multi factor decision-making method to make a comprehensive evaluation of things affected by many factors. Its characteristic is that the evaluation results are not absolutely positive or negative, but represented by a fuzzy set.

1. **Modeling and Analysis**

通过一定的数据分析和处理, 利用机器学习中的lgb模型对数据进行建模, 最终得到定量的数学模式帮助阳光公司分析市场, 最终获得一定的成功.

Through certain data analysis and processing, the LGB model in machine learning is used to model the data, and the quantitative mathematical model is obtained to help Sunshine Company analyze the market, and finally achieve certain success

**5.1 Data Preprocessing**

The title presents three annexes, which respectively provide data on the three categories of hair dryer, dryer and pacifier: 11470, 1615 and 18939. Also, the three data sets provided contain product user ratings and reviews extracted from the Amazon Customer Reviews Dataset thru Amazon Simple Storage Service(Amazon S3). Meanwhile, in each attachment, each row of data contains information of 15 categories of the commodity, and each category represents the specific meaning as shown in the following table\*.

Table \*:

|  |  |
| --- | --- |
| marketplace (string) | 2 letter country code of the marketplace where the review was written. |
| customer\_id (string) | Random identifier that can be used to aggregate reviews written by a single author. |
| review\_id (string) | The unique ID of the review. |
| product\_id (string) | The unique Product ID the review pertains to. |
| product\_parent (string) | Random identifier that can be used to aggregate reviews for the same product. |
| product\_title (string) | Title of the product. |
| product\_category (string) | The major consumer category for the product. |
| star\_rating (int) | The 1-5 star rating of the review. |
| helpful\_votes (int) | Number of helpful votes. |
| total\_votes (int) | Number of total votes the review received. |
| vine (string) | Customers are invited to become Amazon Vine Voices based on the trust that they have earned in the Amazon community for writing accurate and insightful reviews. Amazon provides Amazon Vine members with free copies of products that have been submitted to the program by vendors. Amazon doesn't influence the opinions of Amazon Vine members, nor do they modify or edit reviews. |
| verified\_purchase (string) | A “Y” indicates Amazon verified that the person writing the review purchased the product at Amazon and didn't receive the product at a deep discount. |
| review\_headline (string) | The title of the review. |
| review\_body (string) | The review text. |
| review\_date (bigint) | The date the review was written. |

For a large number of commodity evaluation information data, in order to facilitate the subsequent establishment of mathematical model to solve the problem, the data is preprocessed according to the following steps:

Step 1: After checking all the data, it is found that only four rows of data have missing values, which has little impact on the overall data volume, so the four rows of data are selected for deletion.

Step 2: After careful observation of the data, it can be found that the values of all sample data in the two categories of market place and product category are the same, indicating that these two categories are irrelevant variables in this study, so they are deleted; at the same time, in the data in the column of product\_parent, except for the two outliers in the pacifier, all other data, as long as the product\_id is the same, The same is true for product\_parent, so only one field is reserved.

Step 3: Because product\_id and product\_title have the same meaning that both stand for a same product, which leads to the data redundancy, delete product title.

Step 4: Add two columns of data, year and month, according to the data of review\_date.

Step 5: Text processing, such as word segmentation, stop words removal, punctuation removal, is convenient for later text analysis

**5.2 LightGBM Model**

LightGBM是对GBDT的高效实现，加速了传统GBDT训练过程20倍以上，同时达到了几乎相同的精度。主要包括两个核心部分，一个是GOSS--在减少数据量和保证精度上平衡的算法。另一个是EFB--能够将许多互斥的特征变为低维稠密的特征,有效的避免不必要0值特征的计算。

LightGBM is an efficient implementation of GBDT, which accelerates the traditional gbdt training process by more than 20 times, and achieves almost the same accuracy. It mainly includes two core parts, one is GOSS, which is an algorithm to reduce the amount of data and ensure the balance of accuracy, the other is EFB, which can change many mutually exclusive features into low-dimensional dense features, and effectively avoid unnecessary calculation of zero value features.

GOSS算法排除了大部分具有小梯度的数据，只使用剩余的数据进行信息增益估计，LightGBM研究[1]表示:具有较大梯度的样本在计算信息增益的时候扮演着更加重要的角色，GOSS可以通过更加小规模的数据来获得非常精准的信息增益计算。

GOSS algorithm excludes most of the data with small gradients and only uses the remaining data for information gain estimation. LightGBM research [1] shows that: samples with large gradients play a more important role in calculating information gain. GOSS can obtain very accurate information gain calculation through smaller data.

EFB算法通过将互斥的特征捆绑在一起，来减少特征数目。互斥特征意味着它们几乎很少同时出现在非零值，并且LightGBM也表明:找到最优化特征捆绑是NP问题，但是贪心算法能够获得非常好的近似概率

EFB algorithm reduces the number of features by binding mutually exclusive features together. Mutually exclusive features mean that they rarely appear in non-zero values at the same time, and lightGBM also shows that it is NP problem to find the optimal feature binding, but greedy algorithm can obtain very good approximate probability.

**5.2.1 Model Establishmen and analysis**

We will follow the following process to build a model to solve the first question.



Figure 1: the process of building the model

Among them, hair\_dryer, microwave and pacifier are data sets after data preprocessing. In the following part, the above process is described in detail

**5.2.2 Emotional Score**

We use TextBlob model [加参考文献]which is used in Sentiment Analysis to quantify the reviews\_body. The result of TextBlob Sentiment Analysis can return a tuple in the form of(polarity, subjectivity).

Table 2: Emotional analysis results

|  |  |  |
| --- | --- | --- |
| reviews | polarity | subjectivity |
| Works great! | 1.000 | 0.750 |
| Love this dryer! | 0.625 | 0.600 |
| Quiet, but does not seem like 1000 watt power… | 0.000 | 0.333 |
| Amazing addition to the nursery! | 0.750 | 0.900 |
| ⋯ | ⋯ | ⋯ |

The Polarity is a floating point number with a range of [-1.0, 1.0]. Positive Numbers mean positive and negative Numbers mean negative. Alfred is a floating point number with a range of [0.0, 1.0], where 0.0 is objective and 1.0 is subjective. We use the polarity value calculated by reviews to replace the original reviews\_body data, so as to build our model.

After we use TextBlob to map the review body to the numerical space, We build Light gradient accelerator (Light) Gradient Boosting Machine (LightGBM) which is widely used in data mining tasks to get the weights to build model. We take star\_rating as the output and the other features after preprocessing above as the input to train the model.

**5.2.2** **Feature Engineering**

In order to obtain more data correlation information, we used feature engineering to process the data set, and then put it into the model for training. Feature engineering often plays an important role in the field of data mining. Different feature engineering can obtain different feature sets. A good feature engineering can reveal more relevant information in the data set, thus making the model more accurate.

我们观察数据发现，有些评论和评级存在很大的差距性，例如给了好评，却只打了一星，给差评，却打了五星。

Our observation data shows that there is a big gap between some comments and ratings. For example, we only give one star for good comments and five stars for bad comments.

Table 3: Comments on inconformity of star rating and review body

|  |  |  |
| --- | --- | --- |
| Review\_id | Star\_rating | Review\_body |
| R134FUK2D9TQU6 | 1 | I have used the dryer several times and it works great. I had questions which were answered promptly by other customers which was helpful in making my decision. Definitely recommend. |
| R1HI3QGXJQ2RUT | 5 | We owned these from the store and they are exactly the same. Too bad my grandson decided he was done with pacifiers one week later |
| ⋯ | ⋯ | ⋯ |

通过统计发现该类数据在数据集hair\_dryer, microwave, pacifier中分别存在12, 5, 10条,本文认为此类数据的出现为异常情况, 并非正常的评价信息, 故此后的处理中将该类数据进行删除处理.

It is found that there are 12, 5 and 10 pieces of such data in the data set hair player, microwave and pacifier, respectively. This paper considers that the occurrence of such data is abnormal and not normal evaluation information, so this kind of data will be deleted in the post-processing.

同时, 我们根据一个product\_id的所有star\_rating信息进行统计, 利用如下公式计算出一个product\_id的好评率rate:

At the same time, we make statistics according to all the star rating information of a product\_id, and use the following formula to calculate the praise rate of a product \_id:

$$rate=\frac{\sum (star\_rating>=4)}{count\_i}$$

Among them,$count\_i$ represent product\_id for the total number of i.

最后, 我们将数据中的各种字符串类别数据统一为从0开始的连续数字类别数据, 因为接下来的模型需要该形式的类别数据.

Finally, we unify all kinds of string category data in the data into continuous numerical category data starting from 0, because the next model needs this form of category data.

**5.2.2** **Training data construction and Model training**

所有数据经过特征工程处理之后, 训练一个二分类器. 故将评级在4星及以上的视为正例, 4星一下的视为负例. 并且将数据集的20%大小的数据量作为测试数据来评测模型,剩下的数据作为模型的训练数据训练模型.最终得到的数据量大小如下表:

After all the data are processed by feature engineering, a two classifier is trained. Therefore, those with a rating of 4 stars or above are regarded as positive examples, and those with a rating of less than 4 stars are regarded as negative examples. 20% of the data set size is used as the test data to evaluate the model, and the remaining data is used as the training data training model of the model. The final data size is as follows:

Table 4: Data volume of training set and test set

|  |  |  |  |
| --- | --- | --- | --- |
|  | Train data | Test data | Total |
| hair\_dryer | 9163 | 2291 | 11454 |
| microwave | 1288 | 322 | 1610 |
| pacifier | 15130 | 3783 | 18913 |

硬件环境为16G内存+intel core-i78th, 软件平台为windows10. LightGBM模型相关参数如下: num\_leavers=64, learning\_rate=0.09, 然后输入训练数据训练模型.

The hardware environment is 16g memory + Intel core-i78th, and the software platform is windows10. The related parameters of lightGBM model are as follows: num\_learners = 64, learning\_rate = 0.09, and then input the training data to the training model.

**5.2.3 Model Evaluation**

利用测试数据来评价模型, 评价指标为precision, recall,F1值，计算公式如下:

Using test data to evaluate the model, the evaluation index is precision, recall, F1 value, and the calculation formula is as follows:

$$precision=\frac{TP}{TP+FP}$$

$$recall=\frac{TP}{TP+FN}$$

$$F1=\frac{2\*P\*R}{P+R}$$

Among them, $TP$ indicates that the real evaluation results are positive examples and the predicted results of the model are also the total number of positive examples, $FP$ represents that model predicts the total number of negative cases as positive cases, $FN$ represents the total number of positive cases predicted as negative cases.

The test results are shown in Table 5.

Table 5:Test result

|  |  |  |  |
| --- | --- | --- | --- |
| Product | Precision | Recall | F1 |
| hair\_dryer | 0.812 | 0.982 | 0.889 |
| microwave | 0.846 | 0.903 | 0.874 |
| pacifier | 0.892 | 0.989 | 0.939 |

It can be seen from the above table that the F1 value of lightgbm model in three products is high, which proves that the effect of the model is very good.

**5.2.2** **Result Analysis**

上文成功建立了LightGBM模型, 并且我们利用模型可以得出下表数据:

The lightGBM model has been successfully established above, and we can use the model to get the following data:

Table 6: Weight of eigenvalue

|  |  |  |  |
| --- | --- | --- | --- |
|  | Hair\_dryer | Microwave | Pacifier |
| customer\_id | 0.000 | 0.000 | 0.000 |
| review\_id | 0.000 | 0.000 | 0.000 |
| product\_id | 0.036 | 0.000 | 0.009 |
| helpful\_votes | 0.041 | 0.038 | 0.044 |
| total\_votes | 0.138 | 0.122 | 0.139 |
| vine | 0.000 | 0.000 | 0.004 |
| verified\_purchase | 0.028 | 0.028 | 0.025 |
| review\_body | 0.223 | 0.329 | 0.285 |
| review\_date | 0.200 | 0.213 | 0.201 |
| year | 0.003 | 0.000 | 0.000 |
| month | 0.053 | 0.081 | 0.055 |
| rate | 0.274 | 0.184 | 0.233 |

表中数值为各特征的相对权重, 我们可以看出, customer\_id, review\_id在三个数据集上的权重都为0, 说明二者对于阳光公司的市场分析并无帮助, 仅能作为标识符存在, 对结果无影响. 其次我们看到review\_body, rate, review\_date 和total\_votes的权重都相对较高, 证明他们对于一个产品的评级非常重要. 所以, 对于阳光公司而言, 要重点从以上四个方面入手来获得成功.

The values in the table are the relative weights of each feature. We can see that the weights of customer\_ID and review\_ID in the three data sets are all 0, indicating that they are not helpful for the market analysis of sunshine company, only exist as identifiers, and have no impact on the results. Secondly, we can see that the weights of review\_body, rate, review\_date and total\_votes are relatively high, and to prove that they are very important for the rating of a product. Therefore, for sunshine company, we should focus on the above four aspects to achieve success.

**5.3 The Solution for Task A**

一旦阳光公司产品销售, 利用第一题的模型进行预测用户评级. 若预测评级与用户实际评级相符且评级较好, 则通过第一问得出的重要性来分析产品的成功之处, 将成功的方面做到底, 保证产品口碑不下滑. 若评级较差, 同样根据重要性来分析产品在哪些方面做的不够, 接下来应该着重提升产品劣势方面. 实现逆转. 若与预测评级与用户实际评级不符, 则应分析这些用户是否为虚假用户或恶意用户, 从而有针对性的处理该类用户, 保证正常的销售市场.

Once the products of sunshine company are sold, the model of the first question will be used to predict the user rating. If the prediction rating is consistent with the actual rating of the user and the rating is good, the importance of the first question will be used to analyze the success of the products, and the successful aspects will be done to the end to ensure that the product reputation does not decline. If the rating is poor, the importance will also be used to analyze the aspects of the products that are not enough, Next, we should focus on improving the product disadvantage and realizing the reversal. If it is inconsistent with the predicted rating and the actual rating of users, we should analyze whether these users are false users or malicious users, so as to deal with these users in a targeted way and ensure the normal sales market。

**5.4 Task B——Entropy Weight Model**

**5.4.1 Model Building**

针对提供的三个数据集, 本文利用之前的情感分析结果, 结合评级数据, 利用信息熵确定二者权重, 最后得到每个评价的得分, 利用该得分来分析时间模式.

For the three data sets provided, this paper uses the previous results of emotional analysis, combined with rating data, uses information entropy to determine the weight of the two, and finally gets the score of each evaluation, using the score to analyze the time pattern.

信息熵有三个性质，单调性、非负性和累加性：

Information entropy has three properties: monotonicity, nonnegativity and accumulation：

1. 单调性, 发生概率越高的事件, 其携带的信息量越低;

1. Monotonicity: the higher the probability of occurrence, the lower the amount of information it carries;

2.非负性: 信息熵可以看做一种广度量, 非负性是一种合理的必然;

2. Non negativity: information entropy can be regarded as a kind of breadth, and non negativity is a reasonable necessity;

3.累加性: 即多随机事件同时发生存在的总不确定性的量度是可以表示为各事件不确定性的量度的和，这也是广度量的一种体现。

3. Accumulation: that is, the measurement of the total uncertainty of multiple random events occurring at the same time can be expressed as the sum of the measurement of the uncertainty of each event, which is also a reflection of the breadth.

假设用户给商品评级和评价是两个事件和, 期望他们独立, 则有他们同时发生的概率为

Assume that the user's rating and evaluation of goods are the sum of two events, and they are expected to be independent, the probability of their simultaneous occurrence is



According to the accumulation，we can infer that



满足两个变量乘积函数值等于两个变量函数值的和, 应该为对数函数, 再考虑到再考虑到概率都是小于等于1的，取对数之后小于0，考虑到信息熵的第二条性质，所以需要在前边加上负号。最后参考信息论之父克劳德·香农给出的信息熵定义有信息熵公式为:

If the sum of the product function value of two variables is equal to the sum of the function value of two variables, it should be a logarithmic function. Considering that the probability is less than or equal to 1, it is less than 0 after taking the logarithm. Considering the second property of information entropy, it is necessary to add a negative sign in the front. Finally, the definition of information entropy given by Claude Shannon, the father of reference information theory, has the formula of information entropy as follows.



将情感分析后的数据作为评价数据, 将star\_rating作为评级数据, 最终计算产品综合得分score如下公式

Take the data after emotional analysis as evaluation data and star rating as rating data, and finally calculate the product comprehensive score as follows.



emotionScore为对review\_body情感分析的分值, starRating为用户的评级. a和b是利用信息熵公式计算二者的信息熵权值, 对三个数据集分别计算权值得出表xxx:

Emotion score is the score of review\_body sentiment analysis, and starrating is the rating of users. A and B calculate the information entropy weight of the two by using the information entropy formula, and calculate the weight values of the three data sets respectively, as shown in table 7:

Table 7: Information entropy weight

|  |  |  |
| --- | --- | --- |
|  | a | b |
| hair\_dryer | 0.138 | 0.861 |
| microwave | 0.069 | 0.930 |
| pacifier | 0.169 | 0.831 |

Calculate the score of each product to analyze the time-based measurement mode of each data set. In this paper, the overall sales volume, favorable rate, poor rate and other data of each month are calculated by month for visual analysis.单月份整体销量 represents overall sales volume in a single month, indicates favorable rate , indicates bad rate. The calculation formula is as follows:







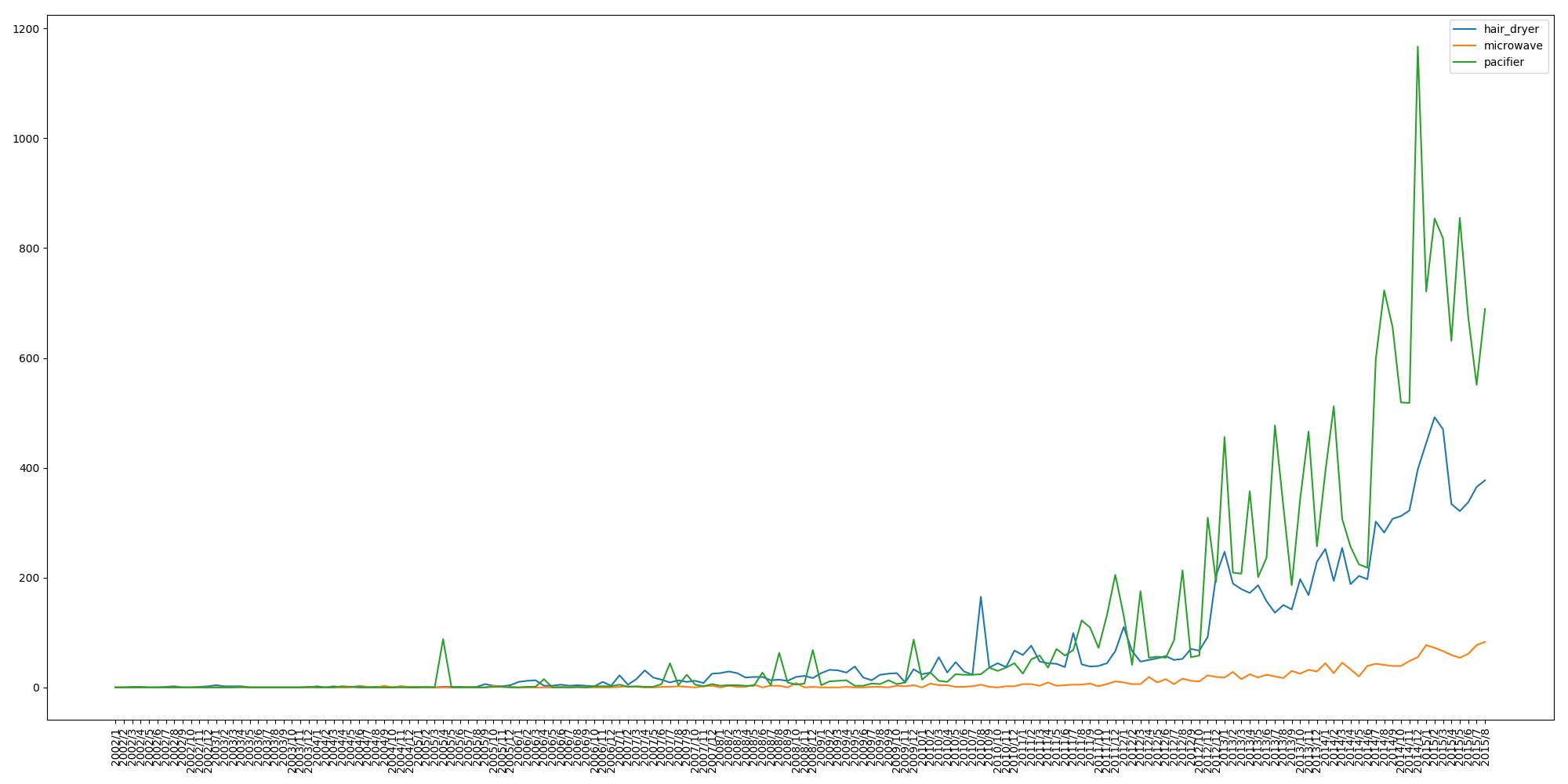
represents a row of data,is in review\_date, andis the score calculated by formula for each piece of data.

**5.4.2 Results**

1. Overall sales volume analysis and data selection

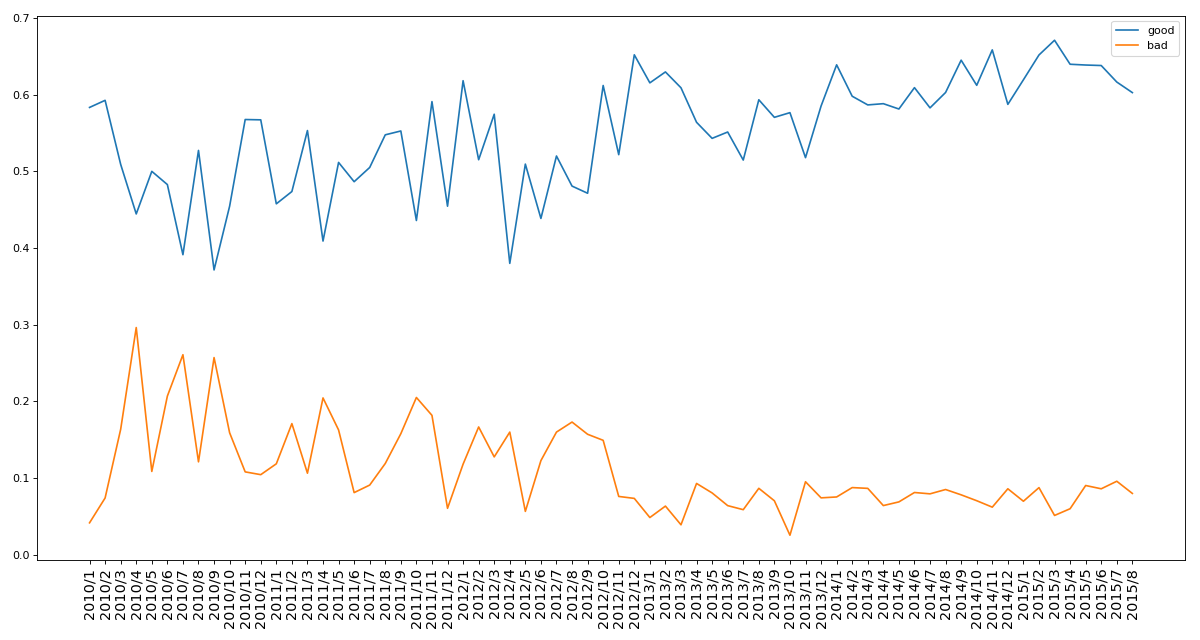
利用预处理的数据, 按月份统计三个产品的销量图如下xxx:

According to the pre-processing data, the sales figures of the three products are as follows:



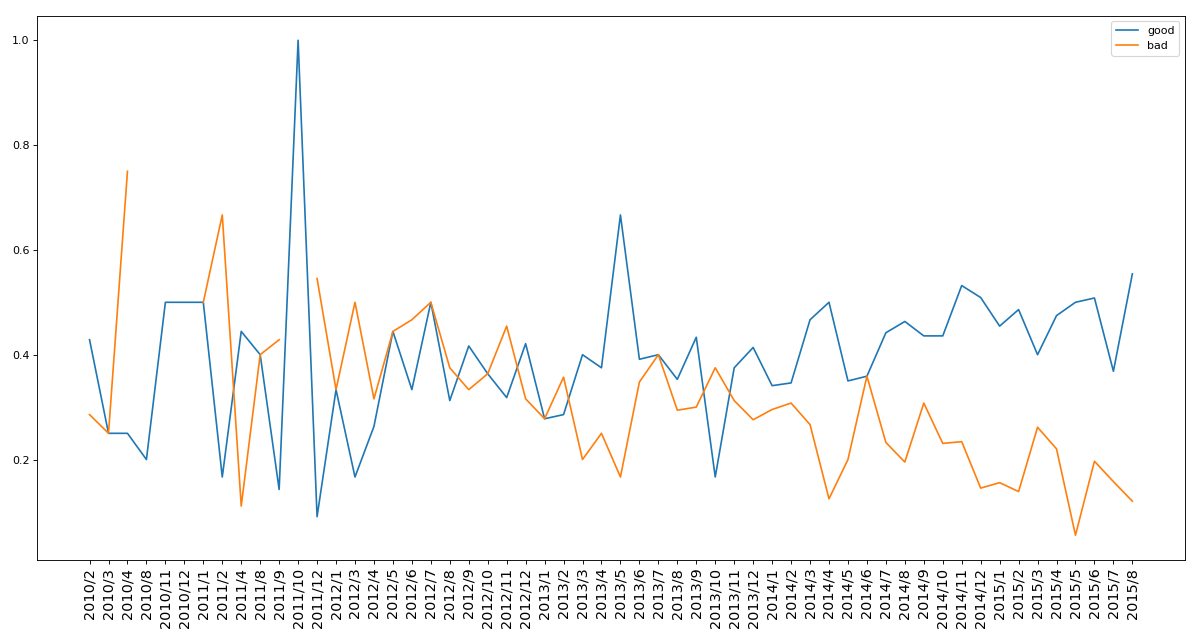
Due to space limitation, image is blurred, and if you want to see the clear version, please refer to the appendix. Through the image, we can know that the sales volume before 2010 is very small. Therefore, the following analysis only takes the data after January 2010 for analysis. Meanwhile, it can be seen that the overall sales volume of the three products is increasing year by year. The sales volume of microwave ovens is relatively low, and the sales volume of diapers is relatively high. The reason may be that the microwave ovens are durable products, while diapers are disposable nondurable products, and the sales volume of hair dryer is in the middle. This is related to the characteristics and uses of the products.

(1)The analysis of hair's time pattern

Using the data after preprocessing, we get the monthly positive and negative rate of hair player as shown in Figure XXX:

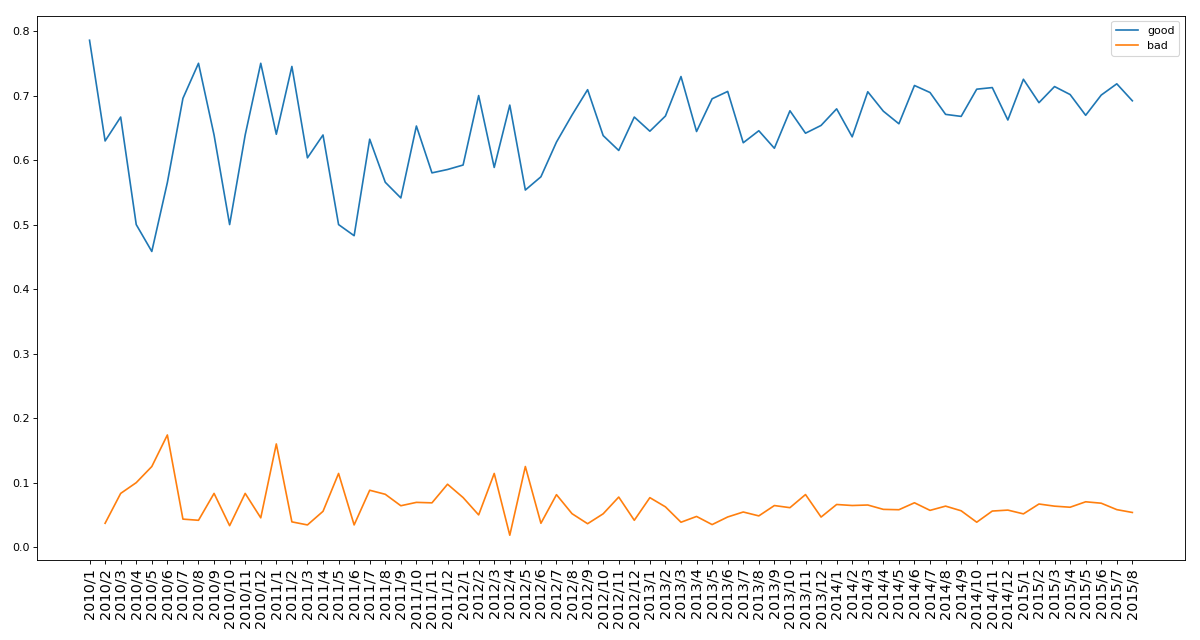
From the figure, we can analyze that since January 2010, the overall positive rate of the hair dryer has increased, and the overall negative rate has declined. And the positive rate has always been far higher than the negative rate. Before January 2013, the negative rate of the whole product is high, and the product quality is not objective, but after 2013, the positive rate of the product is basically high, It shows that the quality of hair dryer gradually rises after market baptism, and people are more comfortable to use it, but there will still be a slight fluctuation

(2) Analysis of time pattern of microwave

Using the data after preprocessing, the monthly positive and negative rates of microwave are as follows:

From the figure, we can see that in the early stage when the microwave oven was put into the market, the market's poor evaluation rate of the microwave oven was slightly higher than the high evaluation rate, and the good evaluation rate of the microwave oven peaked in October 2011, and then declined; from December 2012 to October 2013, the good evaluation rate and bad evaluation rate of the microwave oven fluctuated in a certain range and were relatively stable. In November 2013 and, the good evaluation rate of the product was within a certain range It fluctuates within the range but is always greater than the rate of poor evaluation. Generally speaking, the poor evaluation rate of microwave oven is declining, and the good evaluation rate is rising.

(3) Analysis of pacifier time pattern

Using the data after preprocessing, the monthly positive and negative rates of pacifier are as follows: 

It can be analyzed from the figure that from the time when pacifier put the product into the market to August 2015, the product's favorable rate in the market is far higher than the negative rate, and the fluctuation frequency is getting smaller and smaller, which is in a very stable situation and a very good product.

**5.5 The Solution for Task C**

要求确定一个基于文本的度量方法和基于评级的度量方法的组合, 来指示一个产品的成功或失败. 本文在第一题中建立了lgb模型并得出了相应指标对产品好坏的影响. 同时在上一小问中, 借助熵权法利用评级和评论的情感分值得到了产品的score, 上面两个解决方案都参考了评论和评级, 也即文本和评级, 故此处结合两个解决方案来最好的指示潜在的成功或失败的产品.

It is required to determine the combination of a text-based measurement method and a rating based measurement method to indicate the success or failure of a product. In the first question, the LGB model is established and the impact of the corresponding indicators on the quality of the product is obtained. In the last question, the score of the product is obtained by using the emotional score of rating and comment with the help of entropy weight method, Both of the above solutions refer to comments and ratings, i.e. text and ratings, so the combination of the two solutions best indicates potential successful or failed products.

**5.5.1** **Establishment and Solution of The Model**

结合问题一的特征权重, 进一步筛选出有关文本的和有关评级的特征: review\_body, review\_date, rate, star\_rating. 同时结合上一小问的score. 建立模糊综合评价模型对产品的潜在成功与否进行评价. 模糊综合评价模型的基本步骤如下:

1. 选取特征
2. 确定评价矩阵
3. 检验评价矩阵, 计算权重
4. 计算得分, 得出结果

Combined with the feature weight of question 1, the features of relevant text and rating are further screened: review\_body, review\_date, rate, star\_rating. At the same time, combined with the score of the previous question, the fuzzy comprehensive evaluation model is established to evaluate the potential success of the product. The basic steps of the fuzzy comprehensive evaluation model are as follows:

1. Select features

2. Determine the evaluation matrix

3. Check the evaluation matrix and calculate the weight

4. Calculate the score and get the result

第一步选取特征已经完成. 接下来根据相关资料和数据确定评价矩阵A如下:

The first step is to select features. Next, according to relevant data and data, determine the evaluation matrix A as follows:



通过对举着A进行一致性检验得出CR= 0.018<0.1 故矩阵通过一致性检验. 最后计算矩阵的特征向量并归一化后得到相应特征的权重如下:

Through the consistency test of holding a, it is concluded that CR = 0.018 < 0.1, so the matrix passes the consistency test. Finally, the eigenvector of the matrix is calculated and normalized to obtain the weight of the corresponding features as follows:

Table 8: The weight of the corresponding features

|  |  |
| --- | --- |
| Features | Weights |
| review\_body | 0.158 |
| review\_date | 0.098 |
| rate | 0.259 |
| star\_rating | 0.170 |
| score | 0.315 |

The company can use the weight value to evaluate the success of the product. The larger the value is, the more successful the product is

**5.6 Task D**

我们使用斯皮尔曼模型来测量评级与评论的相关性，斯皮尔曼模型计算最后的结果是一种秩相关系数，它是跟据原始数据的排序进行求解。

We use the Spearman model to measure the correlation between ratings and reviews. The final result of the Spearman model is a rank correlation coefficient, which is solved according to the order of original data.

1.我们将每一条数据根据评论时间进行排序

2.将每一条数据的star\_rating和review的得分情况提取出来作为两个列向量R(Xi)和R(Yi)

3.将R(Xi)和R(Yi)输出到下面公式中：

1. We sort each piece of data according to the comment time;

2. Extract the scores of star rating and review of each data as two column vectors R (XI) and R (Yi);

3. Output R (XI) and R (Yi) to the following formula:

d = \sum_{i=1}^{N} \left | R\left ( X_i\right )-R\left ( Y_i \right ) \right |^{2}

1. 最后，根据下面公式计算出两个列向量之间的相关性Rs,得到结果如表1

4. Finally, the correlation RS between the two column vectors is calculated according to the following formula, and the results are shown in table 9.

Rs = 1-\frac{6 \times d}{N\times\left (N^2 -1 \right )}

Table 9:Coefficient for products

|  |  |
| --- | --- |
| Product | Coefficient |
| hair\_dryer | 0.437 |
| microwave | 0.559 |
| pacifier | 0.372 |

It can be concluded from the table 9 that the correlation coefficient of the three products is greater than 0.05, so specific star ratings will affect the user to write some kind of comment. And we found in the data that some of the comments were indeed affected by the previous comments. And we found in reviews that some of the ratings are as shown in table 10.

Table 10: Partial comment

|  |  |
| --- | --- |
| Review\_id | Reviews |
| R3NQ2GXMX8FDFI | Here's a novel idea!! Read the reviews BEFORE ... |
| |  |  | | --- | --- | | |  | | --- | | RK1L6FAK78AN4 | | | - DO NOT BUY - the 5 star reviews are fake |
| |  | | --- | | ROTACUL0CWK71 | | I don't understand the great reviews for this dryer |
| |  | | --- | | R3EPB70VVPJALX | | Believe the Reviews!!! |
| ⋯ | ⋯ |

It can be seen from the table 10 that the previous rating comments and other information are mentioned in the comments, indicating that some reviews will be affected by other ratings.

**5.7Task E**

题目要求根据文本的评论的特定质量描述符, 和评级相关联, 挖掘其中是否有深层次信息. 为此本文通过对review\_body的具体内容进行挖掘, 提取其相关描述信息, 并通过统计和绘图等方式进行了挖掘.具体步骤如下:

According to the specific quality descriptors of the text comments, the topic is required to be associated with the rating to find out whether there is deep-seated information. Therefore, this paper digs the specific content of the review body, extracts its related description information, and digs it by means of statistics and drawing. The specific steps are as follows:

1. 建立特定的质量描述符词典. 通过查阅资料和相关数据, 本文建立了用户情感-评论-程度词典, 词典中包含英文的各种描述情感、评论词汇、程度词汇等等, 具体词汇表见附录.

1. Establish a specific dictionary of quality descriptors. By consulting materials and relevant data, this paper establishes a user emotion comment degree dictionary, which contains a variety of English description emotions, comment words, degree words and so on. See the appendix for the specific vocabulary

2.将吹风机, 微波炉, 尿不湿数据合并, 同时提取review\_body内容. 按用户评级分类, 分别从review\_body中统计出现在词典中的词语, 通过词频统计方法得到各个词语出现的频次, 将结果按频次降序排列后取出前20个词语. 得到如下表格xxx:

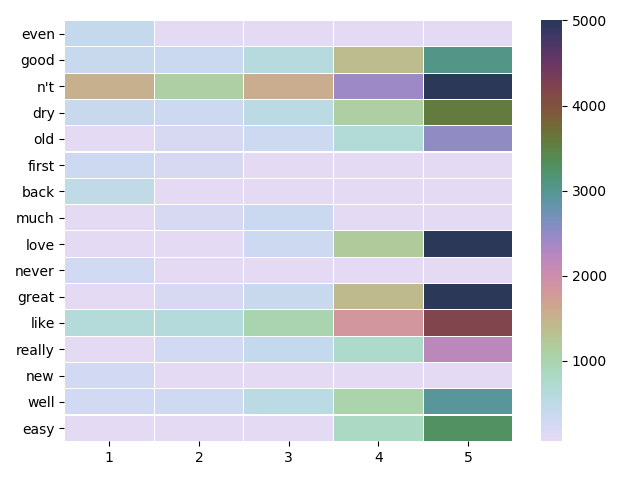
2. Combine the data of hair dryer, microwave and pacifier, extract the content of review body. According to the classification of user rating, count the words appearing in the dictionary from review body, and get the frequency of each word through the word frequency statistical method. Arrange the results in descending order, and then take out the first 20 words. The following table 11 is obtained:

Table 11: Frequency of common words

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 |
| old | 272 | 225 | 353 | 677 | 2512 |
| never | 298 | 0 | 0 | 0 | 0 |
| long | 241 | 186 | 327 | 563 | 1392 |
| new | 286 | 0 | 0 | 0 | 0 |
| good | 392 | 368 | 591 | 1402 | 3058 |
| much | 228 | 234 | 370 | 606 | 2079 |
| back | 460 | 212 | 0 | 0 | 0 |
| bad | 222 | 0 | 0 | 0 | 0 |
| high | 0 | 0 | 232 | 0 | 0 |
| however | 0 | 0 | 265 | 0 | 0 |
| need | 230 | 157 | 252 | 566 | 1645 |
| receive | 222 | 0 | 0 | 0 | 0 |
| keep | 0 | 196 | 261 | 479 | 1695 |
| cute | 0 | 0 | 0 | 0 | 1479 |
| perfect | 0 | 0 | 0 | 0 | 1600 |
| star | 0 | 0 | 0 | 440 | 0 |
| hot | 0 | 187 | 0 | 0 | 0 |
| easy | 0 | 0 | 0 | 832 | 3261 |
| dry | 386 | 326 | 535 | 1108 | 3578 |
| n't | 1509 | 1102 | 1594 | 2445 | 5932 |
| really | 237 | 268 | 417 | 784 | 2209 |
| want | 255 | 157 | 274 | 437 | 0 |
| love | 0 | 159 | 359 | 1184 | 8867 |
| great | 221 | 237 | 388 | 1405 | 5797 |
| nice | 0 | 0 | 0 | 558 | 1405 |
| small | 0 | 165 | 279 | 549 | 0 |
| like | 642 | 602 | 984 | 1851 | 4186 |
| best | 0 | 0 | 0 | 0 | 1448 |
| even | 401 | 208 | 233 | 0 | 0 |
| first | 355 | 223 | 0 | 0 | 1363 |
| still | 228 | 0 | 255 | 506 | 0 |
| set | 0 | 163 | 272 | 607 | 1378 |
| well | 285 | 302 | 531 | 1019 | 2932 |

利用表格数据绘制热力图(图xxx)进行直观分析.

Use table data to draw thermodynamic diagram (Figure XXX) for intuitive analysis.



图中横坐标为星级, 纵坐标为单词, 从图和表中可以看出, 评分为一星的评价出现的单词最多的是” n’t ”是个否定词, 出现了1509次, 接下来的是never, like等; 而评分为5星的出现最多的单词为love(8867次), 其次为great, like等等, 该单词在一星或二星中几乎未出现. 这表明特定的描述词汇与评级是密切相关联的. 同时, 我们也注意到了, “ n’t ”在各个星级中都出现了多次. 通过具体的review\_body数据分析, 在一星级中其多与其他积极词汇同时出现用以表达否定情感, 例如” n’t like ” 等等. 而在五星数据中其常形成许多积极情感的短语, 例如” like n’t allow finger print”等等.

In the figure, the abscissa is the star\_rating, and the ordinate is the word. From the figure and table, it can be seen that "n't" is a negative word, which appears 1509 times in review\_body of five stars , followed by never, like, etc.; while the word with 5 stars is love (8867 times), followed by great, like, etc, These words hardly appears in one or two stars. This shows that specific description words are closely related to rating. At the same time, we also notice that "n't" appears many times in all stars. Through specific review body data analysis, in one star, many of them appear at the same time with other positive words to express negative emotions, For example, "n't like" and so on. In five-star data, it often forms many positive emotional phrases, such as "like n't allow finger print" and so on.

通过上述分析, 我们总结出如下的星级常用词汇表xxx:

Based on the above analysis, we summarize the following common star vocabulary XXX:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 |
| not like | not good | not like | like | love |
| never | not like | not good | well | great |
| back | old | much | great | easy |
| even | not well | old | not | like |
| bad | dry | however | need | well |

* 1. **Task F**

Dear Sir/Madam,

We are very honored to have this opportunity to show you and your company the results of our mining and analysis of the data of hair\_dryer, microwave and pacifier. Here we will give you a comprehensive report about our research in the data.

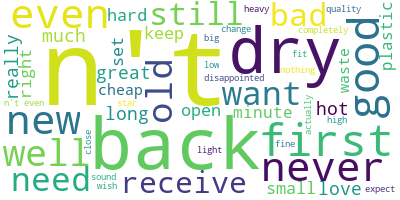
First of all, through the preliminary observations provide three data sets, we find that each product in the marketplace in the US, and product\_category respectively within each data set is the same, we think it useless for analysis, so to be deleted. Similarly, found product\_parent data with product\_id data redundancy, then we delete the product\_parent column data. At the same time, do other data preprocessing.

After emotional analysis in natural language processing, we will not quantitative reviews is mapped to the numerical space, using the numerical range of [1, 1] to comment contains specific emotions. The higher the number, the stronger the positive emotion. After the emotional analysis, we carried out feature engineering. A good feature engineering can make the model better grasp the data features, better discover the internal patterns in the data, and improve the accuracy of the model. Through feature engineering, a new feature rating rate was established and some abnormal conditions were removed. For example, a 1-star comment: "I have used the dryer several times and it works great. I had questions which were answered by other customers which was helpful in making my decision. Definitely recommend.". However, this should be a 5-star favorable rating, so the data with opposite star rating and rating were deleted. Fortunately, that's not a lot of data, with only 27 of the three data sets. Then, we divided the data into the training set and the test set according to the ratio of 8:2, and took the ones with a rating greater than or equal to 4 stars as the positive example, and the ones with a rating less than 4 stars as the negative example, and used the machine learning algorithm LightGBM to conduct separate data modeling for each data set. Finally, the F1 value of the model on the three data sets was 0.889, 0.874 and 0.939, which showed that our model was very successful! Not only that, we also learned from the model that review\_body, review\_date, rate and total\_votes are the most important to the model. Your company should mainly focus on the changes of these four types of data, and try to pay little or no attention to other data with low importance such as reviewers. Once your company plans to sell the three products online, you can use the above model to evaluate the sales data of the products, and compare the predicted results of the model with the actual scores of users to get the result of whether the two are consistent. If it is consistent, it will seize the evaluation characteristics of such users and make targeted improvements to the product to achieve better sales. If it is inconsistent, you can focus on whether such users are fake or malicious users and so on.

We also use entropy weight method to obtain product scores from comprehensive analysis of rating and evaluation, which can measure product reputation and other relevant information. In order to evaluate the potential success or failure of a product, we combined the results of the established machine learning model LighGBM and entropy weight analysis, selected important indicators related to text and rating for fuzzy comprehensive evaluation modeling, and finally concluded that the potential success of a product can be indicated by calculating specific scores.

Finally, we use the correlation analysis technique to obtain that certain ratings will trigger certain types of reviews, that is, users' reviews may be influenced by previous ratings. For example, "-do NOT BUY - the 5 star reviews are fake".

At the same time, we set up a specific vocabulary list and drew the following two word cloud maps for the word frequency statistics of all comment data. The font size of the words in the word cloud map is related to how often they appear in the comments. The higher the frequency, the larger the font. We can see from the word cloud map that the one-star word cloud map is mostly negative words, such as “n't” “like”, “never”, “bad”, while the five-star word cloud map is mostly ‘great’, ‘love’, ‘well’, ‘perfect’, etc. Although there is “n't” in the word cloud image of five stars, we find that it mainly represents positive emotions through specific analysis, such as “like n't allow finger print” and so on.

One star word cloud map Five star word cloud map

Our data mining results are presented, I believe you will have a lot of harvest after you see, finally, I hope your company's products can be a great success!

1. **Conclusion**

通过数据挖掘和深入分析, 我们对数据不仅有了大量的感性认识, 更有很多的理性认识. 我们发现很多时候,, 考察一个产品不能仅仅从对产品的评级来看, 评价, 评论日期等数据也十分重要. 并且通过对三个产品的时间模式进行统计分析得到了产品的销售量, 好评率和差评率随时间的变化模式. 发现pacifier随时间的推迟, 好评越来越多, 好评率比较稳定. 而其他两种产品虽然好评率在上升, 但是仍存在波动的趋势, 发展不够稳定, 尤其是microwave, 一度出现差评率大于好评率的现象, 说明市场上的microwave不能达到人们的满意需求, 阳光公司应着重注意其表现, 保证产品质量的同时注意市场变化情况, 对不同的市场环境进行有针对性的救护措施保证公司效益.

Through data mining and in-depth analysis, we not only have a lot of perceptual knowledge about the data, but also a lot of rational knowledge. We find that many times, it is very important to investigate a product not only from the perspective of product star\_rating, evaluation, review date and other data. And through the statistical analysis of the time patterns of the three products, we get the sales volume of the product, It is found that with the delay of time, pacifier has more and more favorable comments, and the favorable rate is relatively stable. Although the favorable rate of the other two products is rising, there is still a trend of fluctuation, and the development is not stable, especially the phenomenon that the negative rate is greater than the favorable rate once appeared in microave, It shows that the market of microwave can not meet people's needs. Sunshine company should pay attention to its performance, ensure product quality and pay attention to market changes, and take targeted rescue measures for different market environment to ensure the company's benefits  
 利用情感分析技术分析评价中的情感, 再与评级进行比较, 我们发现有五星差评和一星好评等异常情况, 同时, 对评价是否影响评级以及评级是否影响评价进行分析发现, 他们之间互相影响, 好的评级会对接下来的评论产生一定的影响, 同理, 不同的评论短语往往和不同评级对应. 所以在实际分析中, 如果二者出现了不同的趋势, 证明市场可能存在某种不合理的因素, 这时公司需要针对性的调查和解决, 最终使产品良好发展.

Using the emotion analysis technology to analyze the emotion in the review\_body, and then comparing with the rating, we found that there are five-star-poor rating and one star-high praise and other abnormal situations. At the same time, we found that whether the evaluation affects the rating and whether the rating affects the evaluation. They affect each other, and a good rating will have a certain impact on the next comment, Different comment phrases are often corresponding to different ratings. Therefore, in the actual analysis, if there are different trends between the two, it can prove that there may be some unreasonable factors in the market. At this time, the company needs to investigate and solve them, and finally make the product develop well.

1. **Evaluating the Model**

**7.1** **Advantages of the model**

1.我们对评论进行了量化处理，使用了TextBlob进行情感分析对每条评论进行打分，客观地得出了评论对产品的评级。

1. We use TextBlob to analyze the emotion of each review\_body, and score them, so objectively get the rating of the product.

2.在解决问题d的时候，我们使用了斯皮尔曼系数分析，这种分析的要求不需要正态分布，并且它可以得出秩系数，即可以找出变量与变量以某种排序变化的关系。

2. In solving problem D, we use the Spearman coefficient analysis, which requires no normal distribution, and it can get the rank coefficient, that is, it can find out the relationship between variables and variables in a certain order.

3.模型较为客观的利用了尽可能多的数据, 并根据实际数据进行了调整, 通过对数据的可视化分析, 能够直观的挖掘出数据的大量信息, 透过信息来分析三个产品的趋势,声誉,成功与否等具体状况.

4. The model objectively uses as much data as possible, and adjusts it according to the real data. Through the visual analysis of the data, it can directly mine out a large amount of information of the data, and analyze the trend, reputation, success and other specific conditions of the three products through the information above.

**7.2** **Disadvantages of the model**

1.在构建LightGBM模型时, 由于时间问题未能很好的调参, 最终的结果虽然已经够好, 但是可能未达到最好.

1. When building LightGBM model, due to the time limits, the final result is good enough, but may not reach the best.

2.模糊综合评价法的评价矩阵可能不够客观, 但是已经尽量通过数据保证了客观性.

2. The evaluation matrix of fuzzy comprehensive evaluation method may not be objective, but it has guaranteed the objectivity through data as far as possible.

3.词频统计时, 未能很好的将一些短语分析出来, 例如实际上想将”n’t like”等整体表现情感的短语分词, 但是未能很好的做到这一点, 导致最后的评级相关词中展现的不够完整

3. In terms of word frequency statistics, some phrases are not well analyzed, for example, phrases like "n't like" that express emotion through all words are actually segmented, but this is not well done, resulting in incomplete presentation.

**7.3** **Application of the model**

1.模型可以应用在对产品初期的评估，分析其是否可以热销。

2.模型可以分析产品的声誉的变化，公司可以及时调整产品的销售策略。

3.模型可以较为客观的评价产品的成功与否, 并且及时向公司反馈相关信息.

1. The model can be used to evaluate the initial stage of the product and analyze whether it can be sold well.

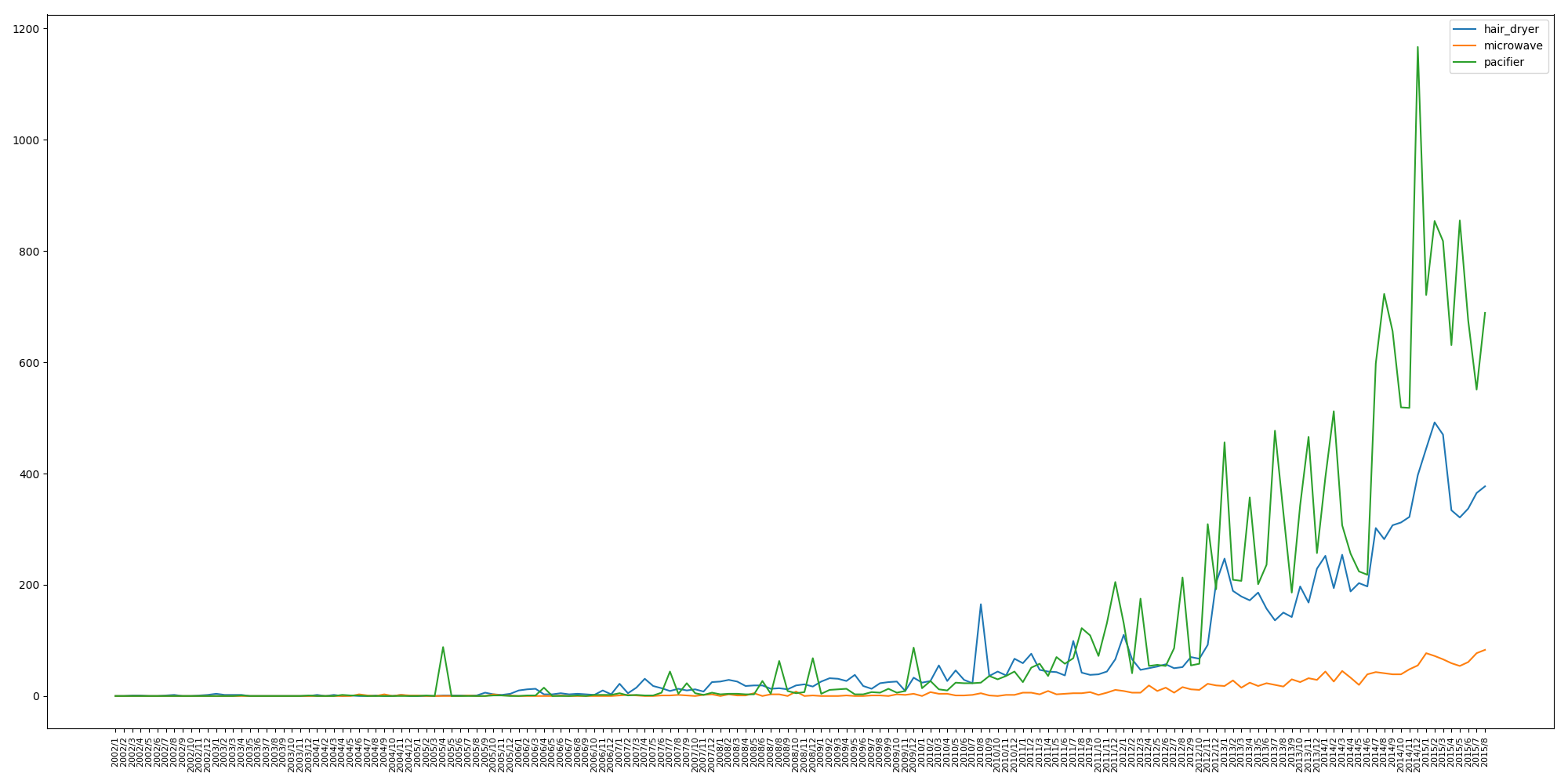
2. The model can analyze the change of product reputation, and the company can adjust the product sales strategy in time.

3. The model can objectively evaluate the success of the product and feed back relevant information to the company in time

1. **Reference**

[1][KE G, MENG Q, FINLEY T, et al. LightCBM: a highly efficient gradient boosting decision tree [C]// Proceedings of the 2017 Annual Conference on Neural Information Processing Systems. New York: Curran Associates Inc., 2017: 3146 -3154.]

Appendix :



|  |  |  |  |
| --- | --- | --- | --- |
| code | anylisis | remark | the code of data anylisis |
| import pandas as pd  import matplotlib.pyplot as plt  from tqdm import tqdm  from my\_util import pre\_process  def null\_process():  hair\_dryer=pd.read\_csv('../Data/hair\_dryer.tsv',sep='\t',encoding='utf-8')  microwave=pd.read\_csv('../Data/microwave.tsv',sep='\t',encoding='utf-8')  pacifier=pd.read\_csv('../Data/pacifier.tsv',sep='\t',encoding='utf-8')  # print(hair\_dryer.head())  for column in hair\_dryer.columns:  if len(set(hair\_dryer[column]))==1:  print('hair\_dryer:',column)  if len(set(microwave[column]))==1:  print('microwave:',column)  if len(set(pacifier[column]))==1:  print('pacifier:',column)  print(set(pacifier['product\_category']))  print(set(pacifier['marketplace']))  print(set(microwave['product\_category']))  print(set(microwave['marketplace']))  #'product\_category', 'marketplace' ,  hair\_dryer['review\_date'] = pd.to\_datetime (hair\_dryer['review\_date'], format='%m/%d/%Y')  hair\_dryer['year'] = hair\_dryer['review\_date'].dt.year  hair\_dryer['month'] = hair\_dryer['review\_date'].dt.month  pacifier['review\_date'] = pd.to\_datetime (pacifier['review\_date'], format='%m/%d/%Y')  pacifier['year'] = pacifier['review\_date'].dt.year  pacifier['month'] = pacifier['review\_date'].dt.month  microwave['review\_date'] = pd.to\_datetime (microwave['review\_date'], format='%m/%d/%Y')  microwave['year'] = microwave['review\_date'].dt.year  microwave['month'] = microwave['review\_date'].dt.month  del hair\_dryer['product\_category']  del hair\_dryer['marketplace']  del pacifier['product\_category']  del pacifier['marketplace']  del microwave['product\_category']  del microwave['marketplace']  del hair\_dryer['product\_title']  del pacifier['product\_title']  del microwave['product\_title']  tmp1 = pacifier[pacifier['product\_id'] == 'b0042i2bwg']  print (tmp1)  tmp2 = pacifier[pacifier['product\_id'] == 'b00db5f114']  print (tmp2)  dic1 = {}  for idx in hair\_dryer.index:  i = hair\_dryer.loc[idx, 'product\_id']  j = hair\_dryer.loc[idx, 'product\_parent']  if i not in dic1:  dic1[i] = [j]  else:  dic1[i].append (j)  for i in dic1:  if len (set (dic1[i])) != 1:  print ('hair\_dryer')  dic2 = {}  for idx in microwave.index:  i = microwave.loc[idx, 'product\_id']  j = microwave.loc[idx, 'product\_parent']  if i not in dic2:  dic2[i] = [j]  else:  dic2[i].append (j)  for i in dic2:  if len (set (dic2[i])) != 1:  print ('microwave')  dic3 = {}  for idx in pacifier.index:  i = pacifier.loc[idx, 'product\_id']  j = pacifier.loc[idx, 'product\_parent']  if i not in dic3:  dic3[i] = [j]  else:  dic3[i].append (j)  for i in dic3:  if len (set (dic3[i])) != 1:  print (i, dic3[i])  print ('pacifier')  # pacifier , , product\_id , product\_parent  del hair\_dryer['product\_parent']  del microwave['product\_parent']  del pacifier['product\_parent']  print(hair\_dryer['product\_id'].count()) #11470  print(microwave['product\_id'].count())#1615  print(pacifier['product\_id'].count())#18939  hair\_dryer=hair\_dryer.dropna()  microwave=microwave.dropna()  pacifier=pacifier.dropna()  print(hair\_dryer['product\_id'].count())#11468  print(microwave['product\_id'].count())#1615  print(pacifier['product\_id'].count())#18937  # , ,  reviewer\_body = []  for i in tqdm (hair\_dryer['review\_body'].values):  sent = ''  for j in pre\_process (i):  sent = sent + ' ' + j  reviewer\_body.append (sent)  hair\_dryer['review\_body'] = reviewer\_body  reviewer\_body = []  for i in tqdm (microwave['review\_body'].values):  sent = ''  for j in pre\_process (i):  sent = sent + ' ' + j  reviewer\_body.append (sent)  microwave['review\_body'] = reviewer\_body  reviewer\_body = []  for i in tqdm (pacifier['review\_body'].values):  sent = ''  try:  for j in pre\_process (i):  sent = sent + ' ' + j  except:  print (i)  sent = i  reviewer\_body.append (sent)  pacifier['review\_body'] = reviewer\_body  hair\_dryer = hair\_dryer.dropna ()  microwave = microwave.dropna ()  pacifier = pacifier.dropna ()  print (hair\_dryer['product\_id'].count ()) # 11468  print (microwave['product\_id'].count ()) # 1615  print (pacifier['product\_id'].count ()) # 18937  hair\_dryer.to\_csv('../Data/hair\_dryer.csv',encoding='utf-8',index=None)  microwave.to\_csv('../Data/microwave.csv',encoding='utf-8',index=None)  pacifier.to\_csv('../Data/pacifier.csv',encoding='utf-8',index=None)  #  # print(hair\_dryer[hair\_dryer.isnull().values==True])  # print(microwave[microwave.isnull().values==True])  # print(pacifier[pacifier.isnull().values==True])  # null\_process()  hair\_dryer=pd.read\_csv('../Data/hair\_dryer.csv',encoding='utf-8')  microwave=pd.read\_csv('../Data/microwave.csv',encoding='utf-8')  pacifier=pd.read\_csv('../Data/pacifier.csv',encoding='utf-8')  print(hair\_dryer.columns)  def fig\_star\_rating\_count():  tmp1=hair\_dryer.groupby(by='star\_rating').count()['customer\_id']  plt.subplot(221)  plt.bar(tmp1.index.values,tmp1.values)  plt.ylim(0,8000)  for a, b in zip(tmp1.index.values, tmp1.values):  plt.text(a, b, '%.0f' % b, ha='center', va='bottom', fontsize=8)  plt.title('hair\_dryer')  tmp2=microwave.groupby(by='star\_rating').count()['customer\_id']  plt.subplot(222)  plt.ylim(0,800)  plt.bar(tmp2.index.values,tmp2.values)  for a, b in zip(tmp2.index.values, tmp2.values):  plt.text(a, b, '%.0f' % b, ha='center', va='bottom', fontsize=8)  plt.title('microwave')  tmp3=pacifier.groupby(by='star\_rating').count()['customer\_id']  plt.subplot(212)  plt.ylim(0,14000)  plt.bar(tmp3.index.values,tmp3.values)  for a, b in zip(tmp3.index.values, tmp3.values):  plt.text(a, b, '%.0f' % b, ha='center', va='bottom', fontsize=8)  plt.title('pacifier')  plt.show()  # fig\_star\_rating\_count()  def fig\_time():  # print(hair\_dryer.groupby('product\_id').count()['customer\_id'].describe())  # hair\_dryer1=hair\_dryer[hair\_dryer['review\_date']>pd.to\_datetime('1/1/2013',format='%m/%d/%Y')]  y1=hair\_dryer.groupby(['year','month']).count()['customer\_id']  y2 = microwave.groupby (['year','month']).count ()['customer\_id']  y3 = pacifier.groupby (['year','month']).count ()['customer\_id']  x=[]  for i in range(2002,2016):  for j in range(1,13):  x.append((i,j))  x.pop(-1)  x.pop(-1)  x.pop(-1)  x.pop(-1)  tmp=[]  for i in x:  if i in list(y1.index.values):  tmp.append(y1.loc[i])  else:  tmp.append(0)  y1=tmp  tmp = []  for i in x:  if i in list(y2.index.values):  tmp.append (y2[i])  else:  tmp.append (0)  y2=tmp  tmp = []  for i in x:  if i in list(y3.index.values):  tmp.append (y3[i])  else:  tmp.append (0)  y3=tmp  x=[str(item[0])+'/'+str(item[1]) for item in x]  plt.figure(figsize=(20,10))  plt.plot(x,y1)  plt.plot(x,y2)  plt.plot(x,y3)  plt.xticks (size='small', rotation=90, fontsize=8)  plt.legend(['hair\_dryer','microwave','pacifier'],loc = 'best')  plt.show ()  print (tmp)  fig\_time()  # test1=hair\_dryer[hair\_dryer['product\_id']=='B003V264WW']  # print()  # print(hair\_dryer.info())  # print(microwave.info())  # print(pacifier.info())  #  # print(hair\_dryer.describe())  # print(microwave.describe())  # print(pacifier.describe())  # # : , , helpful\_votes/verified\_purchase ,  # print(hair\_dryer[hair\_dryer['verified\_purchase']=='Y'].count())  print() | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| code | T1 | remark | | the code of question 1 |
| from textblob import TextBlob  #  # text = "five stars".replace('.','')  # blob = TextBlob (text)  # #  # print ("blob ")  # print (blob)  # print (blob.sentiment)  from sklearn.model\_selection import train\_test\_split  import pandas as pd  import lightgbm as lgb  from tqdm import tqdm  import numpy as np  cat\_cols = ['customer\_id', 'review\_id', 'product\_id', 'vine', 'verified\_purchase']  def pre\_process(prod,data):  del data['review\_headline']  dtime = pd.to\_datetime (data['review\_date'])  v = (dtime.values - np.datetime64 ('2000-01-01T08:00:00Z')) / np.timedelta64 (1, 'ms')  data['review\_date'] = v  data[cat\_cols].astype('category')  def map\_value(x):  x\_set=set(x.values)  dic={}  n=0  for i in x\_set:  if i not in dic:  dic[i]=n  n+=1  new\_x=[]  for i in x.values:  new\_x.append(dic[i])  return new\_x  for col in cat\_cols:  data[col] = map\_value(data[col])  def get\_sentment(col):  new\_col=[]  # for i in tqdm(col):  # out\_put = emotion\_eng.getMoodValue(i)  # new\_col.append(out\_put['all\_value'])  for i in tqdm(col):  out\_put = TextBlob (i)  new\_col.append(out\_put.sentiment.polarity)  return new\_col  data['review\_body']=get\_sentment(data['review\_body'])  def anylisis(data):  not\_pair = data[((data['star\_rating'] == 1) & (data['review\_body'] > 0.6)) | ((data['star\_rating'] == 5) & (data['review\_body'] < -0.6))]  # print(not\_pair)  return not\_pair.index.values  abnormal\_product = {}  abnormal\_product[prod] = (list (anylisis (data))) # 8  print(abnormal\_product)  # abnormal\_product['microwave'] = (list (anylisis (microwave))) # 3  # abnormal\_product['pacifier'] = (list (anylisis (pacifier))) # 18  data = data[~data.index.isin (abnormal\_product[prod])]  return data  hair\_dryer=pd.read\_csv('../Data/hair\_dryer.csv',encoding='utf-8')  hair\_dryer=hair\_dryer.dropna()  hair\_dryer=pre\_process('hair\_dryer',hair\_dryer)  hair\_dryer.to\_csv('../Data/new\_hair\_dryer.csv')  microwave=pd.read\_csv('../Data/microwave.csv',encoding='utf-8')  microwave=microwave.dropna()  microwave=pre\_process('microwave',microwave)  microwave.to\_csv('../Data/new\_microwave.csv')  pacifier=pd.read\_csv('../Data/pacifier.csv',encoding='utf-8')  pacifier=pacifier.dropna()  pacifier=pre\_process('pacifier',pacifier)  pacifier.to\_csv('../Data/new\_pacifier.csv')  def get\_X\_y(prod,data):  cols\_x=['customer\_id', 'review\_id', 'product\_id', 'helpful\_votes', 'total\_votes', 'vine', 'verified\_purchase','review\_body', 'review\_date', 'year', 'month','rate']  star=[]  for i in data['star\_rating']:  if i<4:  star.append(0)  else:  star.append(1)  data['star\_rating']=star  X=data[cols\_x]  X[cat\_cols]=X[cat\_cols].astype('category')  # X['customer\_id']=X['customer\_id'].astype('category')  y=data['star\_rating']  # min\_max\_scaler = MinMaxScaler ()  # X= min\_max\_scaler.fit\_transform (X)  # da=pd.DataFrame(X,columns=cols\_x)  return X,y  def gen\_rate(data):  tmp = data.groupby ('product\_id').count ()['customer\_id']  sums = {}  for i in tmp.index.values:  sums[i] = tmp[i]  rate = {}  for i in sums:  cnt = data[(data['product\_id'] == i) & (data['star\_rating'] < 4)].count ()[0]  rate[i] = cnt / sums[i]  rates = []  for i in data['product\_id'].values:  rates.append (rate[i])  data['rate'] = rates  return data  hair\_dryer=pd.read\_csv('../Data/new\_hair\_dryer.csv',encoding='utf-8',index\_col=0)  microwave=pd.read\_csv('../Data/new\_microwave.csv',encoding='utf-8',index\_col=0)  pacifier=pd.read\_csv('../Data/new\_pacifier.csv',encoding='utf-8',index\_col=0)  hair\_dryer=gen\_rate(hair\_dryer)  microwave=gen\_rate(microwave)  pacifier=gen\_rate(pacifier)  def model1():  X,y=get\_X\_y('hair\_dryer',hair\_dryer)  print(len(y))  X\_train, X\_test, y\_train, y\_test = train\_test\_split(X,y, test\_size=0.2, random\_state=0,shuffle=True)  print("Train data length:", len(X\_train))  print("Test data length:", len(X\_test))  print(' !')  # cv and train  gbm = lgb.sklearn.LGBMClassifier(boosting\_type='gbdt', num\_leaves=64, max\_depth=-1, learning\_rate=0.09, n\_estimators=10, max\_bin=255, subsample\_for\_bin=200000, objective=None, min\_split\_gain=0.0, min\_child\_weight=0.001, min\_child\_samples=20, subsample=1.0, subsample\_freq=1, colsample\_bytree=1.0, reg\_alpha=0.0, reg\_lambda=0.0, random\_state=None, n\_jobs=-1, silent=True)  gbm.fit(X\_train,y\_train,sample\_weight=None, init\_score=None,  eval\_set=None, eval\_names=None, eval\_sample\_weight=None,  eval\_class\_weight=None, eval\_init\_score=None, eval\_metric=None,  early\_stopping\_rounds=None, verbose=True,  feature\_name='auto', categorical\_feature='auto', callbacks=None)  print ('Start predicting...')  #  y\_pred = gbm.predict (X\_test)  from sklearn.metrics import precision\_score, recall\_score, roc\_auc\_score  print('importance:',list(zip(X\_train.columns.values,gbm.feature\_importances\_)))  precision=precision\_score(y\_test, y\_pred)  recall=recall\_score(y\_test, y\_pred)  print(list(zip(y\_test.values,y\_pred)))  print (' ', precision)  print (' ', recall)  print ('auc ', roc\_auc\_score (y\_test, y\_pred))  print ('F1 ', 2 \* (precision \* recall) / (precision + recall))  def model2():  X, y = get\_X\_y ('microwave',microwave)  print (len (y))  X\_train, X\_test, y\_train, y\_test = train\_test\_split (X, y, test\_size=0.2, random\_state=0, shuffle=True)  print ("Train data length:", len (X\_train))  print ("Test data length:", len (X\_test))  print (' !')  # cv and train  gbm = lgb.sklearn.LGBMClassifier (boosting\_type='gbdt', num\_leaves=64, max\_depth=-1, learning\_rate=0.1,  n\_estimators=10, max\_bin=255, subsample\_for\_bin=200000, objective=None,  min\_split\_gain=0.0, min\_child\_weight=0.001, min\_child\_samples=20, subsample=1.0,  subsample\_freq=1, colsample\_bytree=1.0, reg\_alpha=0.0, reg\_lambda=0.0,  random\_state=None, n\_jobs=-1, silent=True)  gbm.fit (X\_train, y\_train, sample\_weight=None, init\_score=None,  eval\_set=None, eval\_names=None, eval\_sample\_weight=None,  eval\_class\_weight=None, eval\_init\_score=None, eval\_metric=None,  early\_stopping\_rounds=None, verbose=True,  feature\_name='auto', categorical\_feature='auto', callbacks=None)  print ('Start predicting...')  #  y\_pred = gbm.predict (X\_test)  from sklearn.metrics import precision\_score, recall\_score, roc\_auc\_score  print ('importance:', list (zip (X\_train.columns.values, gbm.feature\_importances\_)))  precision = precision\_score (y\_test, y\_pred)  recall = recall\_score (y\_test, y\_pred)  print (list (zip (y\_test.values, y\_pred)))  print (' ', precision)  print (' ', recall)  print ('auc ', roc\_auc\_score (y\_test, y\_pred))  print ('F1 ', 2 \* (precision \* recall) / (precision + recall))  def model3():  X, y = get\_X\_y ('pacifier',pacifier)  print (len (y))  X\_train, X\_test, y\_train, y\_test = train\_test\_split (X, y, test\_size=0.2, random\_state=0, shuffle=True)  print ("Train data length:", len (X\_train))  print ("Test data length:", len (X\_test))  print (' !')    gbm = lgb.sklearn.LGBMClassifier (boosting\_type='gbdt', num\_leaves=64, max\_depth=-1, learning\_rate=0.09,  n\_estimators=10, max\_bin=255, subsample\_for\_bin=200000, objective=None,  min\_split\_gain=0.0, min\_child\_weight=0.001, min\_child\_samples=20, subsample=1.0,  subsample\_freq=1, colsample\_bytree=1.0, reg\_alpha=0.0, reg\_lambda=0.0,  random\_state=None, n\_jobs=-1, silent=True)  gbm.fit (X\_train, y\_train, sample\_weight=None, init\_score=None,  eval\_set=None, eval\_names=None, eval\_sample\_weight=None,  eval\_class\_weight=None, eval\_init\_score=None, eval\_metric=None,  early\_stopping\_rounds=None, verbose=True,  feature\_name='auto', categorical\_feature='auto', callbacks=None)  print ('Start predicting...')  #  y\_pred = gbm.predict (X\_test)  from sklearn.metrics import precision\_score, recall\_score, roc\_auc\_score  print ('importance:', list (zip (X\_train.columns.values, gbm.feature\_importances\_)))  precision = precision\_score (y\_test, y\_pred)  recall = recall\_score (y\_test, y\_pred)  print (list (zip (y\_test.values, y\_pred)))  print (' ', precision)  print (' ', recall)  print ('auc ', roc\_auc\_score (y\_test, y\_pred))  print ('F1 ', 2 \* (precision \* recall) / (precision + recall))  # hair\_dryer  model1()  #microwave  model2()  #pacifier  model3() | | | | |
| code | T2\_a | | remark | the code of question 2-a |
| from sklearn.decomposition import pca  from sklearn.preprocessing import StandardScaler, MinMaxScaler  import numpy as np  import pandas as pd  import math  from textblob import TextBlob  # blob = TextBlob ("text")  # print(blob.sentiment.polarity)  # out\_put = emotion\_eng.getMoodValue("great")#out\_put['all\_value']  # all\_low=hair\_dryer[(hair\_dryer['star\_rating']<2) & (hair\_dryer['review\_body']<-0.6)]  # all\_high=hair\_dryer[(hair\_dryer['star\_rating']>3) & (hair\_dryer['review\_body']>0.2)]  # all\_mid=hair\_dryer[(hair\_dryer['star\_rating']>=2) & (hair\_dryer['star\_rating']<=3) & (0.2>=hair\_dryer['review\_body']) & (hair\_dryer['review\_body']>=-0.6)]  # not\_pair=hair\_dryer[((hair\_dryer['star\_rating']<2) & (hair\_dryer['review\_body']>0.8)) | ((microwave['star\_rating']==5) & (hair\_dryer['review\_body']<-0.6))]  #  # a=all\_low.count()['star\_rating']  # b=all\_high.count()['star\_rating']  # c=all\_mid.count()['star\_rating']  # d=hair\_dryer.count()['star\_rating']  # e=not\_pair.count()['star\_rating']  # print(a,b,c,e,d-a-b-c)  # print()  # all\_low=microwave[(microwave['star\_rating']<2) & (microwave['review\_body']<-0.6)]  # all\_high=microwave[(microwave['star\_rating']>3) & (microwave['review\_body']>0.2)]  # all\_mid=microwave[(microwave['star\_rating']>=2) & (microwave['star\_rating']<=3) & (0.2>=microwave['review\_body']) & (microwave['review\_body']>=-0.6)]  # not\_pair=microwave[((microwave['star\_rating']<2) & (microwave['review\_body']>0.8)) | ((microwave['star\_rating']==5) & (microwave['review\_body']<-0.6))]  #  # a=all\_low.count()['star\_rating']  # b=all\_high.count()['star\_rating']  # c=all\_mid.count()['star\_rating']  # d=microwave.count()['star\_rating']  # e=not\_pair.count()['star\_rating']  # print(a,b,c,e,d-a-b-c)  # print()  # all\_low=pacifier[(pacifier['star\_rating']<2) & (pacifier['review\_body']<-0.6)]  # all\_high=pacifier[(pacifier['star\_rating']>3) & (pacifier['review\_body']>0.2)]  # all\_mid=pacifier[(pacifier['star\_rating']>=2) & (pacifier['star\_rating']<=3) & (0.2>=pacifier['review\_body']) & (pacifier['review\_body']>=-0.6)]  #  # not\_pair=pacifier[((pacifier['star\_rating']<2) & (pacifier['review\_body']>0.8)) | ((pacifier['star\_rating']==5) & (pacifier['review\_body']<-0.6))]  # a=all\_low.count()['star\_rating']  # b=all\_high.count()['star\_rating']  # c=all\_mid.count()['star\_rating']  # d=pacifier.count()['star\_rating']  # e=not\_pair.count()['star\_rating']  # print(a,b,c,e,d-a-b-c)  hair\_dryer=pd.read\_csv('../Data/new\_hair\_dryer.csv',encoding='utf-8',index\_col=0)  microwave=pd.read\_csv('../Data/new\_microwave.csv',encoding='utf-8',index\_col=0)  pacifier=pd.read\_csv('../Data/new\_pacifier.csv',encoding='utf-8',index\_col=0)  def anylisis(data):  all\_low=data[(data['star\_rating']<2) & (data['review\_body']<-0.6)]  all\_high=data[(data['star\_rating']>3) & (data['review\_body']>0.2)]  all\_mid=data[(data['star\_rating']>=2) & (data['star\_rating']<=3) & (0.2>=data['review\_body']) & (data['review\_body']>=-0.6)]  # not\_pair 1 5  not\_pair=data[((data['star\_rating']==1) & (data['review\_body']>0.6)) | ((data['star\_rating']==5) & (data['review\_body']<-0.6))]  a=all\_low.count()['star\_rating']  b=all\_high.count()['star\_rating']  c=all\_mid.count()['star\_rating']  d=data.count()['star\_rating']  e=not\_pair.count()['star\_rating']  # print(a,b,c,e,d-a-b-c)  # print(' 1 5 :',e)  return not\_pair.index.values  abnormal\_product={}  abnormal\_product['hair\_dryer']=(list(anylisis(hair\_dryer)))#8  abnormal\_product['microwave']=(list(anylisis(microwave)))#3  abnormal\_product['pacifier']=(list(anylisis(pacifier)))#18  print(abnormal\_product)  def scaler(X):  """    """  min\_max\_scaler = MinMaxScaler ()  x\_train= min\_max\_scaler.fit\_transform (X)  x=pd.DataFrame(x\_train,columns=X.columns.values)  return x  def cal\_weight(x):  ''' '''  #  x = x.apply (lambda x: ((x - np.min (x)) / (np.max (x) - np.min (x))))  # k  rows = x.index.size #  cols = x.columns.size #  k = 1.0 / math.log (rows)  lnf = [[None] \* cols for i in range (rows)]  # --  #  # p=array(p)  x = np.array (x)  lnf = [[None] \* cols for i in range (rows)]  lnf = np.array (lnf)  for i in range (0, rows):  for j in range (0, cols):  if x[i][j] == 0:  lnfij = 0.0  else:  p = x[i][j] / x.sum (axis=0)[j]  lnfij = math.log (p) \* p \* (-k)  lnf[i][j] = lnfij  lnf = pd.DataFrame (lnf)  E = lnf  #  d = 1 - E.sum (axis=0)  #  w = [[None] \* 1 for i in range (cols)]  for j in range (0, cols):  wj = d[j] / sum (d)  w[j] = wj  # ,  w = pd.DataFrame (w)  return w  def get\_eval(prod,data):  data=data[~data['product\_id'].isin(abnormal\_product[prod])]  x=data[['star\_rating','review\_body']]  # x=scaler(x)  w = cal\_weight (x) # cal\_weight  w.index = x.columns  w.columns = ['weight']  wei={'star\_rating':w.loc['star\_rating','weight'],'review\_body':w.loc['review\_body','weight']}  return wei  # wei=get\_eval('hair\_dryer',hair\_dryer) #{'star\_rating': 0.8529774897515476, 'review\_body': 0.1470225102484523}  # print(wei)  def gen\_score(prod,data):  x=data['star\_rating'].values  y=data['review\_body'].values  wei = get\_eval (prod, data)  print(prod,wei)  score=np.array(x)\*wei['star\_rating']+np.array(y)\*wei['review\_body']  data['score']=score  return data  data1=gen\_score('hair\_dryer',hair\_dryer)  data2=gen\_score('microwave',microwave)  data3=gen\_score('pacifier',pacifier)  print(data1.head())  print(data2.head())  print(data3.head()) | | | | |
| code | T2\_b | | remark | the code of question 2-b |
| from sklearn.decomposition import pca  from sklearn.model\_selection import train\_test\_split  from sklearn.preprocessing import StandardScaler, MinMaxScaler  import numpy as np  import pandas as pd  import math  import matplotlib.pyplot as plt  from sklearn.preprocessing import StandardScaler  from sklearn.linear\_model import LogisticRegression  from sklearn.linear\_model import SGDClassifier  from sklearn.metrics import classification\_report  from textblob import TextBlob  # blob = TextBlob ("text")  # print(blob.sentiment.polarity)  # out\_put = emotion\_eng.getMoodValue("great")#out\_put['all\_value']  # all\_low=hair\_dryer[(hair\_dryer['star\_rating']<2) & (hair\_dryer['review\_body']<-0.6)]  # all\_high=hair\_dryer[(hair\_dryer['star\_rating']>3) & (hair\_dryer['review\_body']>0.2)]  # all\_mid=hair\_dryer[(hair\_dryer['star\_rating']>=2) & (hair\_dryer['star\_rating']<=3) & (0.2>=hair\_dryer['review\_body']) & (hair\_dryer['review\_body']>=-0.6)]  # not\_pair=hair\_dryer[((hair\_dryer['star\_rating']<2) & (hair\_dryer['review\_body']>0.8)) | ((microwave['star\_rating']==5) & (hair\_dryer['review\_body']<-0.6))]  #  # a=all\_low.count()['star\_rating']  # b=all\_high.count()['star\_rating']  # c=all\_mid.count()['star\_rating']  # d=hair\_dryer.count()['star\_rating']  # e=not\_pair.count()['star\_rating']  # print(a,b,c,e,d-a-b-c)  # print()  # all\_low=microwave[(microwave['star\_rating']<2) & (microwave['review\_body']<-0.6)]  # all\_high=microwave[(microwave['star\_rating']>3) & (microwave['review\_body']>0.2)]  # all\_mid=microwave[(microwave['star\_rating']>=2) & (microwave['star\_rating']<=3) & (0.2>=microwave['review\_body']) & (microwave['review\_body']>=-0.6)]  # not\_pair=microwave[((microwave['star\_rating']<2) & (microwave['review\_body']>0.8)) | ((microwave['star\_rating']==5) & (microwave['review\_body']<-0.6))]  #  # a=all\_low.count()['star\_rating']  # b=all\_high.count()['star\_rating']  # c=all\_mid.count()['star\_rating']  # d=microwave.count()['star\_rating']  # e=not\_pair.count()['star\_rating']  # print(a,b,c,e,d-a-b-c)  # print()  # all\_low=pacifier[(pacifier['star\_rating']<2) & (pacifier['review\_body']<-0.6)]  # all\_high=pacifier[(pacifier['star\_rating']>3) & (pacifier['review\_body']>0.2)]  # all\_mid=pacifier[(pacifier['star\_rating']>=2) & (pacifier['star\_rating']<=3) & (0.2>=pacifier['review\_body']) & (pacifier['review\_body']>=-0.6)]  #  # not\_pair=pacifier[((pacifier['star\_rating']<2) & (pacifier['review\_body']>0.8)) | ((pacifier['star\_rating']==5) & (pacifier['review\_body']<-0.6))]  # a=all\_low.count()['star\_rating']  # b=all\_high.count()['star\_rating']  # c=all\_mid.count()['star\_rating']  # d=pacifier.count()['star\_rating']  # e=not\_pair.count()['star\_rating']  # print(a,b,c,e,d-a-b-c)  hair\_dryer=pd.read\_csv('../Data/new\_hair\_dryer.csv',encoding='utf-8',index\_col=0)  microwave=pd.read\_csv('../Data/new\_microwave.csv',encoding='utf-8',index\_col=0)  pacifier=pd.read\_csv('../Data/new\_pacifier.csv',encoding='utf-8',index\_col=0)  def gen\_rate(data):  tmp = data.groupby ('product\_id').count ()['customer\_id']  sums = {}  for i in tmp.index.values:  sums[i] = tmp[i]  rate = {}  for i in sums:  cnt = data[(data['product\_id'] == i) & (data['star\_rating'] < 4)].count ()[0]  rate[i] = cnt / sums[i]  rates = []  for i in data['product\_id'].values:  rates.append (rate[i])  data['rate'] = rates  return data  hair\_dryer=gen\_rate(hair\_dryer)  microwave=gen\_rate(microwave)  pacifier=gen\_rate(pacifier)  def anylisis(data):  all\_low=data[(data['star\_rating']<2) & (data['review\_body']<-0.6)]  all\_high=data[(data['star\_rating']>3) & (data['review\_body']>0.2)]  all\_mid=data[(data['star\_rating']>=2) & (data['star\_rating']<=3) & (0.2>=data['review\_body']) & (data['review\_body']>=-0.6)]  # not\_pair 1 5  not\_pair=data[((data['star\_rating']==1) & (data['review\_body']>0.6)) | ((data['star\_rating']==5) & (data['review\_body']<-0.6))]  a=all\_low.count()['star\_rating']  b=all\_high.count()['star\_rating']  c=all\_mid.count()['star\_rating']  d=data.count()['star\_rating']  e=not\_pair.count()['star\_rating']  # print(a,b,c,e,d-a-b-c)  # print(' 1 5 :',e)  return not\_pair.index.values  abnormal\_product={}  abnormal\_product['hair\_dryer']=(list(anylisis(hair\_dryer)))#8  abnormal\_product['microwave']=(list(anylisis(microwave)))#3  abnormal\_product['pacifier']=(list(anylisis(pacifier)))#18  print(abnormal\_product)  def scaler(X):  """    """  min\_max\_scaler = MinMaxScaler ()  x\_train= min\_max\_scaler.fit\_transform (X)  x=pd.DataFrame(x\_train,columns=X.columns.values)  return x  def cal\_weight(x):  ''' '''  #  x = x.apply (lambda x: ((x - np.min (x)) / (np.max (x) - np.min (x))))  # k  rows = x.index.size #  cols = x.columns.size #  k = 1.0 / math.log (rows)  lnf = [[None] \* cols for i in range (rows)]  # --  #  # p=array(p)  x = np.array (x)  lnf = [[None] \* cols for i in range (rows)]  lnf = np.array (lnf)  for i in range (0, rows):  for j in range (0, cols):  if x[i][j] == 0:  lnfij = 0.0  else:  p = x[i][j] / x.sum (axis=0)[j]  lnfij = math.log (p) \* p \* (-k)  lnf[i][j] = lnfij  lnf = pd.DataFrame (lnf)  E = lnf  #  d = 1 - E.sum (axis=0)  #  w = [[None] \* 1 for i in range (cols)]  for j in range (0, cols):  wj = d[j] / sum (d)  w[j] = wj  # ,  w = pd.DataFrame (w)  return w  def get\_eval(prod,data):  data=data[~data['product\_id'].isin(abnormal\_product[prod])]  x=data[['star\_rating','review\_body']]  # x=scaler(x)  w = cal\_weight (x) # cal\_weight  w.index = x.columns  w.columns = ['weight']  wei={'star\_rating':w.loc['star\_rating','weight'],'review\_body':w.loc['review\_body','weight']}  return wei  # wei=get\_eval('hair\_dryer',hair\_dryer) #{'star\_rating': 0.8529774897515476, 'review\_body': 0.1470225102484523}  # print(wei)  def gen\_score(prod,data):  x=data['star\_rating'].values  y=data['review\_body'].values  wei = get\_eval (prod, data)  score=np.array(x)\*wei['star\_rating']+np.array(y)\*wei['review\_body']  data['score']=score  return data  def fig(prod, D):  data=gen\_score(prod,D)[['review\_date','year','month','score']]  # print(data.describe())  good=data[(data['score']>4) & (data['year']>2009)].groupby(['year','month']).count()['score']  bad=data[(data['score']<1) & (data['year']>2009)].groupby(['year','month']).count()['score']  all\_of=data[(data['year']>2009)].groupby(['year','month']).count()['review\_date']  good=pd.DataFrame(good,index=good.index.values)  bad=pd.DataFrame(bad,index=bad.index.values)  all\_of=pd.DataFrame(all\_of,index=all\_of.index.values)  bad.rename(columns={'score':'score\_bad'},inplace=True)  x=[str(i[0])+'/'+str(i[1]) for i in good.index.values]  good['time']=x  x=[str(i[0])+'/'+str(i[1]) for i in bad.index.values]  bad['time']=x  x=[str(i[0])+'/'+str(i[1]) for i in all\_of.index.values]  all\_of['time']=x  all=pd.merge(good,bad,how='left')  all=pd.merge(all,all\_of,how='left')  all.fillna(0)  fig = plt.figure(num=1, figsize=(15, 8),dpi=80)  plt.plot(all['time'].values,all['score'].values/all['review\_date'].values)  # plt.show()  plt.plot(all['time'].values,all['score\_bad'].values/all['review\_date'].values)  # plt.plot(all['time'].values,all['review\_date'].values)  plt.legend(['good','bad'],loc = 'best')  plt.xticks (size='small', rotation=90, fontsize=13)  plt.show()  # fig('hair\_dryer',hair\_dryer)  # fig('microwave',microwave)  # fig('pacifier',pacifier)  def classify(prod,data):  data=gen\_score(prod, data)  cols\_x = ['helpful\_votes', 'total\_votes', 'verified\_purchase', 'review\_body', 'review\_date', 'month', 'rate','score']  x=data[cols\_x]  scores=data['star\_rating'].values  y=[]  for score in scores:  if score>=4:  y.append(1)  else:  y.append(0)  X\_train, X\_test, y\_train, y\_test = train\_test\_split(x,y,test\_size=0.2,random\_state=0)  ss = StandardScaler ()  X\_train = ss.fit\_transform (X\_train)  X\_test = ss.fit\_transform (X\_test)  lr = LogisticRegression()  lr.fit (X\_train, y\_train)  lr\_y\_predict = lr.predict (X\_test)  print(lr\_y\_predict)  print ('Accuracy of LR Classifier:', lr.score (X\_test, y\_test))  print()  classify('hair\_dryer',hair\_dryer)  print() | | | | |
| code | T2\_c | | remark | the code of question 2-c |
|  | | | | |
| code | T2\_d | | remark | the code of question 2-d |
| import pandas as pd  del\_cols = [&apos;customer\_id&apos;,&apos;review\_id&apos;,&apos;product\_id&apos;,&apos;helpful\_votes&apos;,&apos;total\_votes&apos;,&apos;vine&apos;,&apos;verified\_purchase&apos;,&apos;review\_date&apos;,&apos;year&apos;,&apos;month&apos;]  hair\_dryer=pd.read\_csv(&apos;../Data/new\_hair\_dryer.csv&apos;,encoding=&apos;utf-8&apos;,index\_col=0) hair\_dryer=hair\_dryer.drop(del\_cols,axis=1) microwave=pd.read\_csv(&apos;../Data/new\_microwave.csv&apos;,encoding=&apos;utf-8&apos;,index\_col=0) microwave=microwave.drop(del\_cols,axis=1) pacifier=pd.read\_csv(&apos;../Data/new\_pacifier.csv&apos;,encoding=&apos;utf-8&apos;,index\_col=0) pacifier=pacifier.drop(del\_cols,axis=1) print(&apos;hair\_dryer\n&apos;,hair\_dryer.corr(&apos;spearman&apos;)) print() print(&apos;microwave\n&apos;,microwave.corr(&apos;spearman&apos;)) print() print(&apos;pacifier\n&apos;,pacifier.corr(&apos;spearman&apos;)) print() | | | | |
| code | T2\_e | | remark | the code of question 2-e |
| import collections  import pickle  import numpy as np  import matplotlib.pyplot as plt  import jieba.analyse  import seaborn as sns  from tqdm import tqdm  from my\_util import pre\_process  import pandas as pd  import wordcloud  # TF - IDF jieba.analyse.extract\_tags (sentence, topK=20, withWeight=False, allowPOS=())  # TextRank jieba.analyse.textrank (sentence, topK=20, withWeight=False, allowPOS=('ns', 'n', 'vn', 'v'))  hair\_dryer=pd.read\_csv('../Data/hair\_dryer.csv',encoding='utf-8')  microwave=pd.read\_csv('../Data/microwave.csv',encoding='utf-8')  pacifier=pd.read\_csv('../Data/pacifier.csv',encoding='utf-8')  hair\_dryer = hair\_dryer.dropna ()  microwave = microwave.dropna ()  pacifier = pacifier.dropna ()  def try1():  def gen\_star\_sent(n):  tmp1=hair\_dryer[hair\_dryer['star\_rating']==n]['review\_body']  tmp2=microwave[microwave['star\_rating']==n]['review\_body']  tmp3=pacifier[pacifier['star\_rating']==n]['review\_body']  star\_str=''  for i in tqdm(tmp1.values):  for j in pre\_process(i):  star\_str=star\_str+' '+j  for i in tqdm(tmp2.values):  for j in pre\_process (i):  star\_str = star\_str + ' ' + j  for i in tqdm(tmp3.values):  for j in pre\_process (i):  star\_str = star\_str + ' ' + j  print()  return star\_str  one\_star\_sen=gen\_star\_sent(1)  two\_star\_sen=gen\_star\_sent(2)  three\_star\_sen=gen\_star\_sent(3)  four\_star\_sen=gen\_star\_sent(4)  five\_star\_sen=gen\_star\_sent(5)  # one\_star\_sen=hair\_dryer[hair\_dryer['star\_rating']==1]['review\_body']+microwave[microwave['star\_rating']==1]['review\_body']+pacifier[pacifier['star\_rating']==1]['review\_body']  # keywords=jieba.analyse.extract\_tags(one\_star\_sen, topK=20, withWeight=False, allowPOS=())  # print(keywords)  w = wordcloud.WordCloud(max\_words=50)  w.generate(one\_star\_sen)  w.to\_file('output1.png')  # keywords=jieba.analyse.extract\_tags(two\_star\_sen, topK=20, withWeight=False, allowPOS=())  # print(keywords)  w = wordcloud.WordCloud(max\_words=50)  w.generate(two\_star\_sen)  w.to\_file('output2.png')  keywords=jieba.analyse.extract\_tags(three\_star\_sen, topK=20, withWeight=False, allowPOS=())  print(keywords)  w = wordcloud.WordCloud(max\_words=50)  w.generate(three\_star\_sen)  w.to\_file('output3.png')  # keywords=jieba.analyse.extract\_tags(four\_star\_sen, topK=20, withWeight=False, allowPOS=())  # print(keywords)  w = wordcloud.WordCloud(max\_words=50)  w.generate(four\_star\_sen)  w.to\_file('output4.png')  # keywords=jieba.analyse.extract\_tags(five\_star\_sen, topK=20, withWeight=False, allowPOS=())  # print(keywords)  w = wordcloud.WordCloud(max\_words=50)  w.generate(five\_star\_sen)  w.to\_file('output5.png')  def try2():  # words\_list=set()  # with open('emotion\_dict/words\_list.txt','r',encoding='utf-8') as f:  # for line in f:  # words\_list.add(line.replace('\n',''))  # def gen\_star\_sent(n):  # tmp1 = hair\_dryer[hair\_dryer['star\_rating'] == n]['review\_body']  # tmp2 = microwave[microwave['star\_rating'] == n]['review\_body']  # tmp3 = pacifier[pacifier['star\_rating'] == n]['review\_body']  #  # star\_str = ''  # for i in tqdm(tmp1.values):  # for j in pre\_process(i):  # if j in words\_list:  # star\_str = star\_str + ' ' + j  # for i in tqdm(tmp2.values):  # for j in pre\_process (i):  # if j in words\_list:  # star\_str = star\_str + ' ' + j  # for i in tqdm(tmp3.values):  # for j in pre\_process (i):  # if j in words\_list:  # star\_str = star\_str + ' ' + j  # print()  #  # return star\_str  #  # one\_star\_sen = gen\_star\_sent (1)  # two\_star\_sen = gen\_star\_sent (2)  # three\_star\_sen = gen\_star\_sent (3)  # four\_star\_sen = gen\_star\_sent (4)  # five\_star\_sen = gen\_star\_sent (5)  # star\_sent = {}  # star\_sent['one'] = one\_star\_sen  # star\_sent['two'] = two\_star\_sen  # star\_sent['three'] = three\_star\_sen  # star\_sent['four'] = four\_star\_sen  # star\_sent['five'] = five\_star\_sen  # pickle.dump (star\_sent, open ('star\_sent\_cloud.pkl', 'wb'))  star\_sent=pickle.load(open('star\_sent\_cloud.pkl','rb'))  w = wordcloud.WordCloud (max\_words=50)  w.generate (star\_sent['one'])  w.to\_file ('output1.png')  w = wordcloud.WordCloud (max\_words=50)  w.generate (star\_sent['two'])  w.to\_file ('output2.png')  w = wordcloud.WordCloud (max\_words=50)  w.generate (star\_sent['three'])  w.to\_file ('output3.png')  w = wordcloud.WordCloud (max\_words=50)  w.generate (star\_sent['four'])  w.to\_file ('output4.png')  w = wordcloud.WordCloud (max\_words=50)  w.generate (star\_sent['five'])  w.to\_file ('output5.png')  def try3(num):  words\_list = set ()  with open ('emotion\_dict/words\_list.txt', 'r', encoding='utf-8') as f:  for line in f:  words\_list.add (line.replace ('\n', ''))  def gen\_star\_sent1(n):  tmp1 = hair\_dryer[hair\_dryer['star\_rating'] == n]['review\_body']  tmp2 = microwave[microwave['star\_rating'] == n]['review\_body']  tmp3 = pacifier[pacifier['star\_rating'] == n]['review\_body']  star\_str = []  for i in tqdm (tmp1.values):  for j in pre\_process (i):  if j in words\_list:  star\_str .append(j)  for i in tqdm (tmp2.values):  for j in pre\_process (i):  if j in words\_list:  star\_str .append(j)  for i in tqdm (tmp3.values):  for j in pre\_process (i):  if j in words\_list:  star\_str .append(j)  print()  return star\_str  # one\_star\_sen = gen\_star\_sent1 (1)  # two\_star\_sen = gen\_star\_sent1 (2)  # three\_star\_sen = gen\_star\_sent1 (3)  # four\_star\_sen = gen\_star\_sent1 (4)  five\_star\_sen = gen\_star\_sent1 (5)  # star\_sent={}  # star\_sent['one']=one\_star\_sen  # star\_sent['two']=two\_star\_sen  # star\_sent['three']=three\_star\_sen  # star\_sent['four']=four\_star\_sen  # star\_sent['five']=five\_star\_sen  # pickle.dump(star\_sent,open('star\_sent\_count.pkl','wb'))  star\_sent=pickle.load(open('star\_sent\_count.pkl','rb'))  words\_set=[]  words\_dict={}  word\_counts = collections.Counter (star\_sent['five']) #  word\_counts\_top20 = word\_counts.most\_common (num) # 20  # print (word\_counts\_top20) #  word\_counts\_top20 = dict (word\_counts\_top20)  words\_set += list (word\_counts\_top20.keys ())  words\_dict['5'] = word\_counts\_top20  y5 = list(word\_counts\_top20.values())  x5 = [5 for \_ in range (len (y5))]  word\_counts = collections.Counter (star\_sent['four']) #  word\_counts\_top20 = word\_counts.most\_common (num) # 20  # print (word\_counts\_top20) #  word\_counts\_top20 = dict (word\_counts\_top20)  words\_set += list (word\_counts\_top20.keys ())  words\_dict['4'] = word\_counts\_top20  y4 = list(word\_counts\_top20.values())  x4 = [4 for \_ in range (len (y4))]  word\_counts = collections.Counter (star\_sent['three']) #  word\_counts\_top20 = word\_counts.most\_common (num) # 20  # print (word\_counts\_top20) #  word\_counts\_top20 = dict (word\_counts\_top20)  words\_set += list (word\_counts\_top20.keys ())  words\_dict['3'] = word\_counts\_top20  y3 = list(word\_counts\_top20.values())  x3 = [3 for \_ in range (len (y3))]  word\_counts = collections.Counter (star\_sent['two']) #  word\_counts\_top20 = word\_counts.most\_common (num) # 20  # print (word\_counts\_top20) #  word\_counts\_top20 = dict (word\_counts\_top20)  words\_set += list (word\_counts\_top20.keys ())  words\_dict['2'] = word\_counts\_top20  y2 = list(word\_counts\_top20.values())  x2 = [2 for \_ in range (len (y2))]  word\_counts = collections.Counter (star\_sent['one']) #  word\_counts\_top20 = word\_counts.most\_common (num) # 20  word\_counts\_top20=dict(word\_counts\_top20)  words\_set+=list(word\_counts\_top20.keys())  words\_dict['1']=word\_counts\_top20  # print (word\_counts\_top20) #  y1 = list(word\_counts\_top20.values())  x1=[1 for \_ in range(len(y1))]  words\_set=set(words\_set)  sorted(words\_set)  dic={'1':[],'2':[],'3':[],'4':[],'5':[]}  for i in words\_set:  for j in range(1,6):  if i in words\_dict[str(j)].keys():  dic[str(j)].append(words\_dict[str(j)][i])  else:  dic[str(j)].append(0)  data=pd.DataFrame(dic,index=words\_set)  data.to\_csv('../Data/word\_count'+str(num)+'.csv',encoding='utf-8')  cmap = sns.cubehelix\_palette (start=1.5, rot=3, gamma=0.8, as\_cmap=True)  sns.heatmap (data,linewidths = 0.05, vmax=5000, vmin=50, cmap=cmap)  plt.show()  plt.savefig('words\_hot'+str(num)+'.png')    try3(10)  # try2() | | | | |