Text analysis of Amazon Review

Name: Fang Zhu

Student ID: 2109853WIM200016

January 10, 2022

Abstract

Amazon is one of the largest e-commerce platforms in the world, it is very important for merchants to specify effective sales strategies and product improvement plans based on its massive data. In this project, we perform data analysis and modeling on the rating and review data of T10 Earbuds products. And the establishment of merchant improvements is given based on our results.

In the data analysis module, we first visualized the distribution of ratings, and most of them are five-star positive reviews, which shows that the products are still relatively popular. Then, we analyzed the word frequency and word frequency of the negative review texts with ratings less than 3. We found that users' negative comments mainly focus on the quality control of the product itself, such as wearing comfort and battery life.

In the classification prediction module of the review, we simplified the multi-classification problem into a binary classification problem, and the evaluation metrics shows that the tuned SVM is better than the KNN model. This shows that the SVM is more suitable for the establishment of our rating prediction model. Source code of the project can be found in https://github.com/zhufang9819/must_21opentds.

Contents

1 Introduction

With the rapid development of the Internet, people's shopping methods are no longer limited to offline shopping but are more and more inclined to online shopping. As a seller, whether you can give an effective sales strategy will affect your entire future sales. This project combines big data and artificial intelligence tools to analyze the crawled Amazon product data and provides advice on downstream tasks such as sales and even product design.

The whole project is divided into two parts. First, we preprocessed the original data, and conducted word frequency analysis specifically for the negative evaluation data, and combined the word frequency results to provide solutions that can improve product design. Then, we based on these data and SVM, KNN to build the text classification model. The classification model can get the input review and output whether it is a positive review or a negative review. And these two algorithms are evaluated on the test set.

2 Review Data Analysis

2.1 Overview

Our data is the review data of the T10 Earbuds crawled from the Amazon website[1], with a total of 5507 items. Each item has two columns, the first column is the rating, and the second column is the detailed data of the review. In this section, some statistical and visual analyses will be performed on these data, to mine some suggestions which can improve the sales strategy of the seller. We can use Figure 1

Rate	Review
4	Im one of very , very few people who

Table 1: Sample of our data

and Table 2 to show our rating distribution, in which 5-point reviews occupy the majority, which shows that our product is still very popular. Next, we will start from data cleaning and further analyze the text review data.

[1] https://www.amazon.com/TOZO-Bluetooth-Wireless-Headphones-Waterproof/dp/B07J2Z5DBM/ref=cm_cr_arp_d_bdcrb_top?ie=UTF8&th=1

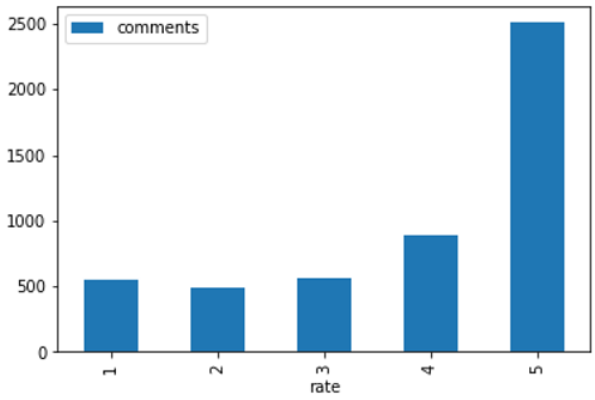

Figure 1: Distribution of our data: Barplot

Rate	1	2	3	4	5
Count	551	491	562	888	2508

Table 2: Distribution of our data: Counting

2.2 Data Cleaning

For some reason, there will be some missing values in our data entries, and these entries with missing values will affect the overall distribution of the data and the analysis results. Therefore, we need to take some strategies to clean these data, because our data only has Two columns, and is complex text data, so we use a strategy of directly discarding rows with missing values to clean the data.

Next, we need to do some preprocessing on the review text. First, we use jieba[2] for word segmentation, then, the result of our word segmentation is converted to lowercase, and then the punctuation and stopwords are removed. The stop word list is selected from the nltk[3] library.

2.3 Negative Review Text Analysis

In order to make their products more popular, merchants should be much more interested in negative reviews than positive reviews. Therefore, this section analyzes the text of negative reviews. First, we filter out negative reviews, that is, reviews

[2] https://github.com/fxsjy/jieba
[3] https://www.nltk.org/

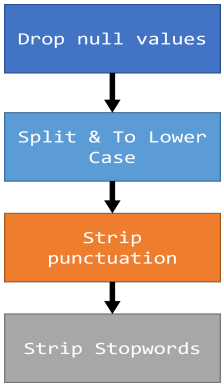

Figure 2: Pipeline of data cleaning

with **rate** < 3 . Then we merge each review, and then perform word frequency analysis on the merged text. The results of the top 50 word frequencies are shown in Figure 3.

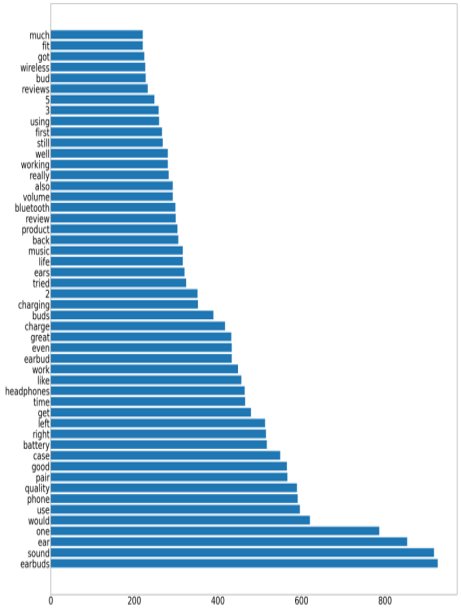

Figure 3: Top 50 of our Negative review word frequency result

Next, we select some high-frequency words related to the product itself for analysis according to the results of word frequency analysis, and give corresponding product improvement suggestions:

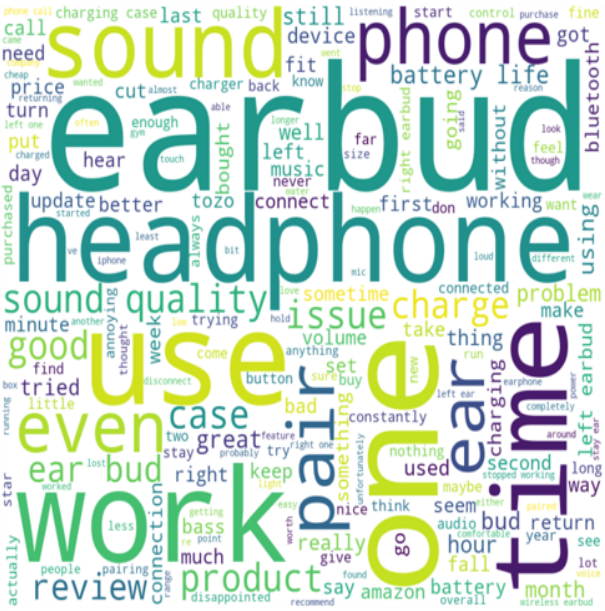

Figure 4: Word Cloud of our Negative review word frequency result

- **volume, sound**: Control the sensitivity of volume adjustment
- **ear**: Improve wearing comfort
- **charge, battery, minute**: Improve the battery quality
- **bluetooth, wireless, connect**: Improve the bluetooth stability.

3 Text Classification Modeling

3.1 Overview

In this section, we train some classifiers on our dataset to classify reviews and evaluate the classifier's performance on the test set as a criterion.

3.2 Problem Simplification

To simplify the problem, we re-divide the ratings of the reviews into two categories.

$$\textit{Negative}\ (\text{Rate} < 3), \quad \textit{Positive}\ (\text{Rate} \geq 3),$$

The first category is the reviews with a rating less than 3, which are called negative reviews, and the second category reviews with a rating greater than or equal to 3, which are called positive reviews. So far, the question becomes a binary classification problem.

3.3 Text Encoding

Because the comment text is composed of strings and it is difficult for the computer to directly process the non-numeric data obtained from the plain text. To fit the model, we need to encode non-numeric objects such as words to convert them into numerical values.

This project uses the **TF-IDF** encoding method[1]. We need to calculate the *word frequency* TF and the *inverse document frequency* IDF. Given the document d_j and the word t_i, the word frequency TF is defined as

$$TF\left(t_i, d_j\right) = \frac{n_{i,j}}{\sum_k n_{k,j}}, \tag{1}$$

where $n_{i,j}$ represents the number of times t_i appears in d_j, and $\sum_k n_{k,j}$ represents the sum of the number of occurrences of all words in d_j. The inverse file frequency IDF is defined as

$$IDF_i = \lg \frac{|D|}{1 + |\{j : t_i \in d_j\}|}, \tag{2}$$

where $|D|$ represents the total number of documents in the corpus, and $|\{j : t_i \in d_j\}|$ represents the number of documents containing the word t_i. Since this value may be 0, we need to smooth it, that is, add 1 to the denominator. Thus

$$TF - IDF_{i,j} = TF_{i,j} \times IDF_i. \tag{3}$$

Suppose we have k words in our corpus, for each sentence s, we can encode it as

$$x_{s,1}, x_{s,2}, \cdots, x_{s,k}, \tag{4}$$

Where $x_{s,j}$ represents the word $TF - IDF_{s,j}$, so that we can implement TF-IDF encode on each comment to get a numerical data matrix.

3.4 Methods

3.4.1 Support Vector Machine

Support vector machine, also known as SVM, is a classifier proposed by Vapnik[3]. The biggest advantage of SVM is that it can be combined with different kinds of

kernel functions to deal with some nonlinear classification problems[4]. In addition, its variants[5] can also be used to solve regression problems.

Given training vectors

$$x_i \in \mathbb{R}^k, k = 1, \cdots, n,$$

and the label $y \in \{-1, 1\}$, our goal is to find $w \in \mathbb{R}^k$ and $b \in \mathbb{R}$ such that the prediction given by is correct for most samples. SVC [2] solves the following problem:

$$\begin{aligned} \min_{w,b,\zeta} \quad & \frac{1}{2} w^T w + C \sum_{i=1}^{n} \zeta_i \\ \text{s.t.} \quad & y_i \left(w^T \phi \left(x_i \right) + b \right) \geq 1 - \zeta_i \\ & \zeta_i \geq 0, i = 1, \ldots, n \end{aligned} \tag{5}$$

Intuitively, were trying to maximize the margin (by minimizing $\|w^T w\|^2 = w^T w$), while incurring a penalty when a sample is misclassified or within the margin boundary. Ideally, the value $y_i(w^T \phi(x_i) + b)$ would be ≥ 1 for all samples, which indicates a perfect prediction. But problems are usually not perfectly separable by a hyperplane, we allow some samples to be at a distance ζ_i from their correct margin boundary. The penalty term C controls the strength of this penalty, and as a result, acts as an inverse regularization parameter. Equation 5 is called the primal problem, and its dual problem is

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2} \alpha^T Q \alpha - e^T \alpha \\ \text{s.t.} \quad & y^T \alpha = 0 \\ & 0 \leq \alpha_i \leq C, i = 1, \ldots, n \end{aligned} \tag{6}$$

where e is the all ones vector, and Q is a $n \times n$ matrix which is positive semidefinite, where ϕ is the kernel. The terms α_i are called the dual coefficients, and they are upper-bounded by C.

Given a sample x, The final decision function is

$$\sum_{i \in SV} y_i \alpha_i K \left(x_i, x \right) + b, \tag{7}$$

and the prediction is the sign of the decision function.

3.4.2 K-Nearest Neighbour

The k nearest neighbor algorithm is simple and intuitive: given a training dataset, for a new input instance, find the k instances closest to the instance in the training

dataset, and if the majority of these k instances belong to a certain class, the input instance will be classified into this class.

Input Given training set

$$T = \{(x_1, y_1), (x_2, y_2), \cdots, (x_N, y_N)\} \tag{8}$$

where $x_i \in \mathcal{X} \subseteq \mathbb{R}^n$ is the feature vector of the instance $y_i \in \mathcal{Y} = \{c_1, c_2, \cdots, c_k\}$ is the class of the instance, $i = 1, 2, \cdots N$ instance feature vector; instance feature vector x.

Output Class y to which instance x belongs.

- According to the given distance metric , find the k points closest to x in the training set T, and the neighborhood of covering these k points is denoted as $N_k(x)$;
- A root in $N_k(x)$ such as a categorical decision rule (e.g. majority vote) decides the category y of x:

$$y = \arg\max_{c_j} \sum_{x_i \in N_k(x)} I(y_i = c_j), \quad i = 1, 2, \cdots, N; j = 1, 2, \cdots, K \tag{9}$$

In Equation 9, I is the indicator function, namely, $I = 1$ when $y_i = c_j$, and $I = 0$ otherwise.

3.5 Parameter Setting and Evaluation

The ratio of our training set to test set is 80% : 20%. To better compare the classification effects of the two models, we use cross-validation to compare the two models. For SVM, we compared the category of the kernel with the C value, where the candidate value of the kernel is $\{'linear', 'rbf'\}$, and the candidate value of the C value is $\{0.01, 0.1, 1, 10\}$. For KNN, we performed cross-validation for the parameter K value, and its candidate values are $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$. We can directly use the GridSearch in sklearn for cross-validation, and the optimal model is used for evaluation on the test set.

As for performance measurements, we choose three different metrics on the classification task:

$$\begin{aligned} \text{Accuracy} &= \frac{TP+TN}{TP+TN+FP+FN} \\ \text{Precision} &= \frac{TP}{TP+FP} \\ \text{Recall} &= \frac{TP}{TP+FN} \\ F1- \text{ score} &= \frac{2\times \text{ Precision } \times \text{ Recall}}{\text{Precision } + \text{ Recall}} \end{aligned} \tag{10}$$

A detailed definition of TP, TN, FP, FN can be found from [6].

3.6 Results

After the two models are cross-validated, the optimal models obtained will be tested on the test set, and the result is in Table 3. The **0** after the indicator represents the result corresponding to the negative sample, and the **1** represents the result corresponding to the positive sample.

Models	Accuracy	Precision(0)	Precision(1)	Recall(0)	Recall (1)	F1-Score(0)	F1-Score(1)
SVM	**0.84**	**0.68**	**0.87**	**0.45**	0.94	**0.54**	**0.90**
KNN	0.81	0.61	0.84	0.30	**0.95**	0.40	0.89

Table 3: Evaluation result of SVM and KNN on Test Set

In most metrics, SVM outperforms KNN and achieves an accuracy of 0.84. However, the ability of this classifier to identify negative reviews is far worse than that of positive reviews. It may be caused by the following reasons. The first reason is that the distribution is unbalanced, which will cause the classifier to learn insufficiently for categories with few samples, resulting in misclassification. The second reason is Insufficient feature extraction of the text, which makes it impossible for us to obtain enough discriminative latent features for the task of text classification.

4 Conclusion

In this project, we conducted text analysis and modeling on our Amazon reviews from upstream data analysis and downstream classification. The distribution of ratings shows that the products we choose are still relatively popular, and most of them are Five-star praise. Through the analysis of the word frequency of the negative review

text and the word cloud, we found that the user's negative reviews mainly focus on the quality control of the product itself, such as wearing comfort and battery life. In the classification prediction task of reviews, we simplify the multi-class problem into a two-class problem, the final evaluation metrics show that the tuned SVM is better than the KNN model. The evaluation results of both models are poor in the negative category. In the future, we can use more complex pre-trained word vectors and more complex models to fit massive data and use big data tools to build a real-time end-to-end product data display and review classification system.

References

[1] Aizawa, A. An information-theoretic perspective of tf–idf measures. *Information Processing & Management 39*, 1 (2003), 45–65.

[2] Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., VanderPlas, J., Joly, A., Holt, B., and Varoquaux, G. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning* (2013), pp. 108–122.

[3] Cortes, C., and Vapnik, V. Support-vector networks. *Machine learning 20*, 3 (1995), 273–297.

[4] Scholkopf, B. The kernel trick for distances. *Advances in neural information processing systems* (2001), 301–307.

[5] Smola, A. J., and Schölkopf, B. A tutorial on support vector regression. *Statistics and computing 14*, 3 (2004), 199–222.

[6] Wikipedia contributors. Confusion matrix — Wikipedia, the free encyclopedia, 2021. [Online; accessed 9-January-2022].