PRODUCT RATING PREDICTION BASED ON REVIEWS FROM AMAZON FOOD

Louis Lu A53307505

Department of Electrical and Computer Engineering, University of California San Diego

ABSTRACT

Predict products' rating based on reviews of Amazon Food. Applied N-gram, TF-IDF, to train, validate and test and find the relating between review text and review rating.

***Index Terms*—** N-gram, TF-IDF, KNN, Logistic Regression, sentiment analysis

1. INTRODUCTION

Online shopping is getting more and more popular now. At first, we can only buy books and small products online. But now, almost everything is selling online. Electronics products, groceries, and foods etc. Even cars such large items can be sold online and delivery to your home. After Covid-19 outbreaks, in order to avoid contacting with people, the trend of buying products online is even more unstoppable.

One advantage of online shopping is that customers can give rating and write reviews for the products they buy. They can give high rating and leave positive comments to the products they are satisfied with. On the other hand, if customers dislike the products, they can give them low rating and point out the drawback of products in their reviews.

Most potential customers will also consider the rating and read the reviews of products before buying products. Therefore, analyzing these rating and reviews is important for the manufactures and retailer to strengthen their products and make more profits. In this assignment, I tried to analyze buyers' reviews on food and predict the rating of products so that manufactures and retailer can get more detailed inside of their customers and customers' reviews.

2. LITERATURE REVIEW

People understand the sentence in a fraction. However, machines cannot process text data in raw form. They understand the text which is broken down into a numerical format. Bag-of-words and TF-IDF are techniques that convert text sentences into numeric format. Bag-of-words model is utilized in document classification where the frequency of each word which is utilized as a feature for training a classifier. This method can extract features from text documents and these features can be using for training machine learning algorithms and natural language processing. TF-IDF which is a feature term is more important if it has a higher frequency in a text, known as Term Frequency (TF); and feature term is less important if it appears in different text documents in a training set, known as Inverse Document Frequency (IDF).It can be successfully used for words filtering in various subject fields, such as text summarization and classification.

People are interested to find positive and negative opinions shared by other users for features of particular product or service. Sentiment Analysis is a good fit in this application. It is the process of detecting positive or negative sentiment in text, feelings and emotions, urgency or not, and intentions, knows what people comment about product, service topic, issue and event, and aims at extracting opinions and sentiments. Since customers express their feelings and their thoughts everywhere, especially online shopping. They write their feelings or their usage experience online to products. Therefore, Sentiment analysis becomes an essential and effective tool to monitor their customers.

3. DATASET

After researching and discussion, I decided to choose Amazon Review - Food category to analyze. This dataset contains 10,994,353 reviews. For the training performance and accuracy, I selected 70,000 reviews from 20,994,353 reviews and spitted them into three potion as the training set, validation set and test set. There are 50000, 10000, 10000 samples in the datasets respectively. All the reviews I extracted are written after date Jan 1, 2010 and the reviewer had been verified as a true buyer.

3.1. Data Schemes

Each records has several features, including overall rating, whether a verified buyer or not, product's ASIN, submission time of review, reviewer's ID, reviewer's name, and review text. The features I are interested in and going to predict are review text and overall rating score. More detail about the features are shown in Table.

3.2. Data Analysis

In this section, I would try to dig out more valuable fact from the data. For example, the rating distribution, rating count of

each year, reviews' length distribution, etc,

3.2.1. *Rating Distribution*

I extracted the rating score distribution of the data to see how people rate the products. The bar chart is shown as Figure 1. I found that most people tend to give their products five points, so most of them are satisfied with the products they bought.

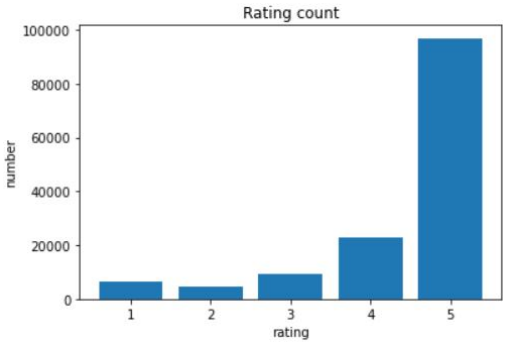

Fig. 1. Rating distribution

3.2.2. *Rating count of each years*

To analyze the recent reviews, I only kept review after Jan 1, 2011. As the Figure 2 shown, most of the reviews I randomly selected from the original data are from 2013 to 2016.

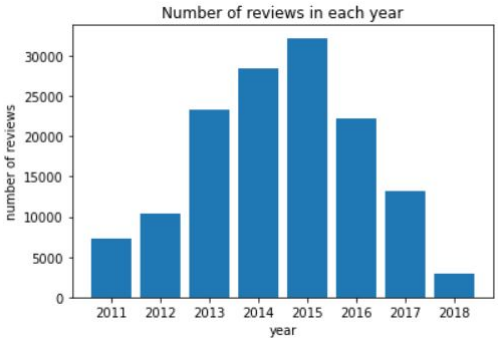

Fig. 2. Number of reviews in each year

3.2.3. *Reviews' length distribution*

In Figure 3 shown, we can found that most reviews' length are between 0 to 500. If I only show length between 0 to 500, as Figure 4. We found that the distribution is presented as the exponential decay.Therefore, most

3.2.4. *Positive and Negative words*

I try to figure out what kinds of words people would use when they gave high rating or low rating. To analyze, I classified 4

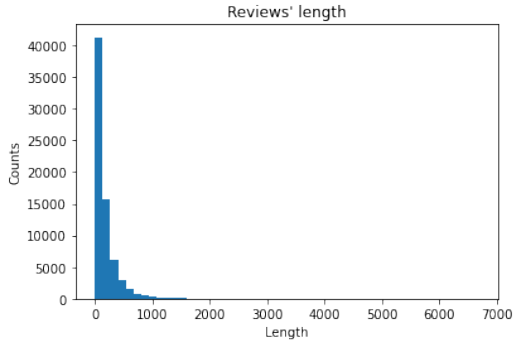

Fig. 3. Reviews' length distribution

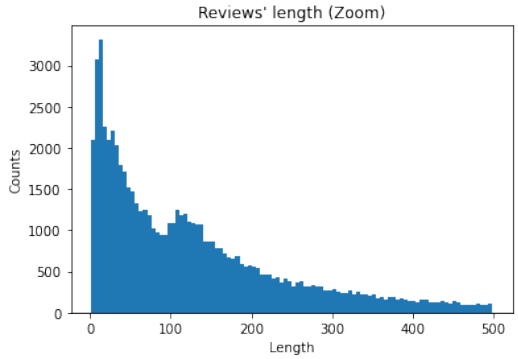

Fig. 4. Reviews' length distribution

and 5 as high rating and 1 to 3 as low rating. Also, I imported Opinion Lexicon which is a dictionary that contains more than 6800 positive and negative words. Therefore, I could focus on positive and negative words and ignore neutral words such as 'the', 'a', etc. Next step, I count the most common words in reviews that are within this dictionary. The result is as follow.

Top 20 sentiment words in positive reviews(above 3): good, like, great, love, best, favorite, well, better, delicious, nice, fresh, hot, sweet, free, recommend, perfect, easy, right, wonderful, strong.

Top 20 sentiment words in negative reviews(below 4): like, good, better, great, love, bad, well, disappointed, sweet, strong, hard, enough, fine, hot, stale, fresh, best, pretty, nice, expensive.

In the review of low rating, it is interesting that not many people use negative words as I expected. Only 'bad' 'disappointed', 'hard', 'stale', 'stale', 'expensive' are in the list of top 20 common words. In fact, a portion of reviewers used 'good' in their reviews. I guess that these reviewers mentioned some good parts of products and then pointed out the drawbacks.

Moreover, In Figure5, we can found that high rating reviews have the frequent word like. In low rating reviews, the frequent words are disappointed, not, awful, okay. I found

that people are more likely to use the words 'if', 'were' and 'was' than the reviews with high rating. The reason might be in these low rating reviews, people wrote about their unhappy experiences during some circumstance, so they used words 'if' or 'when' to describe or assume some situations.

Figure5 is not very clear in this report, the clear image can be found in the code.

4. PREDICTIVE TASK

In this final project, I tried to predict products' rating based on the reviews. I will explain the methods I used to extract features and the methods I used to evaluate the prediction models.

4.1. Feature selection

To extract the features from data for training, I used N-gram and TF-IDF schemes.

4.1.1. N-gram

An n-gram is a contiguous sequence of n items from a given sample of text or speech. The items can be phonemes, syllables, letters, words or base pairs according to the application. The n-grams typically are collected from a text or speech corpus.

In this assignment, we used both unigram scheme and bigram scheme to extract features. All the punctuation and new line characters in the text are removed before extracting the words. The unigram scheme extracts individual word from the text. The implementation of unigram are easy, but the disadvantages are that it loses the information of the word order and the combination of words.

Bigram scheme stores the partial order of words by preserving the 2-word sequence. It try to solve the problem of unigram that missing the order of words, but it also increases the number of entries. With n-gram model, features such as word counts or word frequency can be constructed to help us identify the importance of these n-word sequences.

4.1.2. TF-IDF

TF-IDF is short for term frequency–inverse document frequency, is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus.It is often used as a weighting factor in searches of information retrieval, text mining, and user modeling. It is a combination of test frequency and inverse document frequency, where the two terminologies are defined below.

The definition of TF:

$$td(t,d) = \frac{countoft}{wrodsind} \tag{1}$$

The definition of IDF:

$$idf(t,D) = \log(\frac{N}{|d \in D : t \in d|}) \tag{2}$$

The definition of TFIDF:

$$tfidf(i,d,D) = tf(t,d) \times idf(t,D) \tag{3}$$

The tf–idf value increases proportionally to the number of times a word appears in the document and is offset by the number of documents in the corpus that contain the word, which helps to adjust for the fact that some words appear more frequently in general.

4.2. Evaluating function

Evaluating the algorithms is an essential part of the project. For evaluating the performance of the models, We chose two methods Classification Accuracy and Confusion Matrix.

4.2.1. Classification Accuracy

Classification Accuracy is what we usually mean, when we use the term accuracy. It is the ratio of number of correct predictions to the total number of input samples.

$$Accuracy = \frac{NumberofCorrectPredictions}{TotalNumberofPredictionsMade} \tag{4}$$

4.2.2. Confusion matrix

In the field of machine learning and specifically the problem of statistical classification, a confusion matrix, also known as an error matrix, is a specific table layout that allows visualization of the performance of an algorithm, typically a supervised learning one that I used in this project. Each row of the matrix represents the instances in a predicted class while each column represents the instances in an actual class. The name stems from the fact that it makes it easy to see if the system is confusing two classes.

5. DESCRIPTION OF MODELS

In this assignment, we implemented three classification methods, K-Nearest Neighbors, Logistic Regression, and Support Vector Machine.

5.1. Baseline Model

A naive classifier we used is GaussianNB in sklearn. GaussianNB is a classifier that makes predictions using simple rules, which is useful as a simple baseline to compare with other (real) classifiers. The main idea for this classifier is that we generates predictions by respecting the training set's class distribution. It simply takes the unigram segment and make predictions based on the training set.

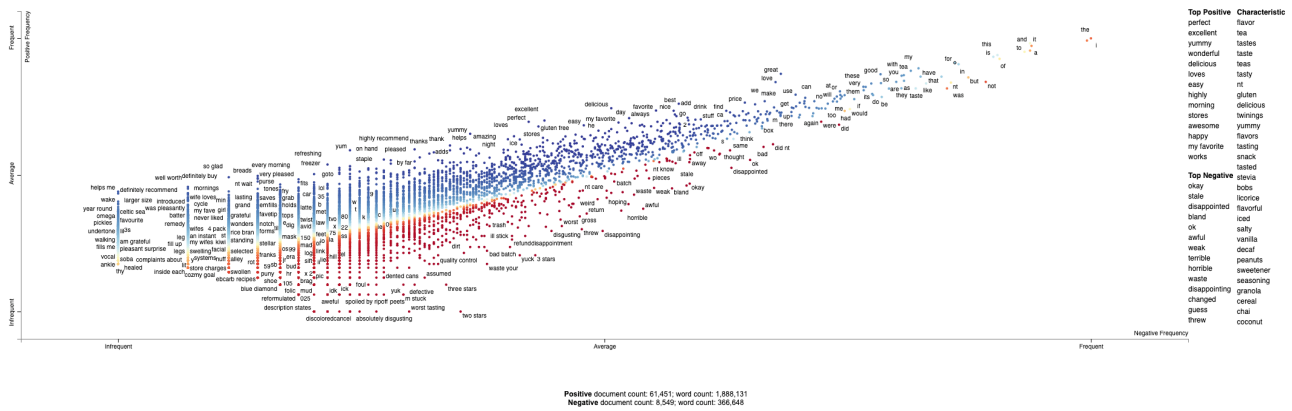

Fig. 5. Scatter plot of positive and negative words

A normal distribution:

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right) \quad (5)$$

5.2. K Nearest Neighbors (KNN)

The first model we implemented is KNN. The KNN algorithm is one of the simplest of all the supervised machine learning algorithms. It simply calculates the distance of a new data point to all other training data points. The distance can be of any type e.g Euclidean or Manhattan etc. It then selects the K-nearest data points, where K can be any integer. Finally the data point is assigned to the class to which the majority of the K data points belong.

For my task, using KNN is straight forward since similar reviews have similar features and they should lead to similar rating on one product. The detailed algorithms is as 2.The benefits of KNN on my task is that as a non-linear classifier, it is good for predictions in complex feature space distribution. However, the original dataset has more than 2,000,000 reviews, we randomly selected 100,000 reviews for training. Otherwise, it took too much time to train the data. Texts are transformed into bag of words and n-gram for analysis in this model.

The disadvantage of KNN is that it doesn't work well with high dimensional data. Because it is more difficult for the algorithm to calculate the distance in each dimension when the number of dimensions is large. Also, the KNN algorithm has a high prediction cost for large datasets, because in large datasets the cost of calculating distance between new point and each existing point becomes higher. Hence, we implemented other models to compare with KNN.

5.3. Logistic Regression

The second model we implemented is Logistic Regression. Logistic regression is a classification algorithm used to assign observations to a discrete set of classes. Logistic regression transforms its output using the logistic sigmoid function to return a probability value.

Logistic Regression is a binary classification model, but the class LogisticRegression in scikit-learn supports multi-class case. In the multiclass case, the training algorithm uses the one-vs-rest (OvR) scheme. It means a class is compared with the rest of the classes. For instance, problem in this assignment is split into 5 binary classification problems:

Problem 1: rating 1 vs [rating 2, 3, 4, 5]
Problem 2: rating 2 vs [rating 1, 3, 4, 5]
Problem 3: rating 3 vs [rating 1, 2, 4, 5]
Problem 4: rating 4 vs [rating 1, 2, 3, 5]
Problem 5: rating 5 vs [rating 1, 2, 3, 4]

After training, validating, and tuning the model, we can get the best result when setting C to 10 and the max iteration number to 1000. The results are shown in the following section and in Table 2.

6. RESULT

In this section, we will show the result of the models. The pipeline for prediction is pre-processing data and extracting features from the data first, then training model using train set, validating using validation set and testing using test set.

6.1. KNN

We can see the performance of KNN classifier on rating product is as Table 2. In the result of this model, we can see that the accuracy of bigram model is sometime slightly larger than unigram scheme regardless of Word bad or TFIDF. And the results were worse in test. The reason might be KNN does not work well with high dimensional data. We can see that KNN tend to predict low rating to rating 5 through confusion matrix in Figure6.

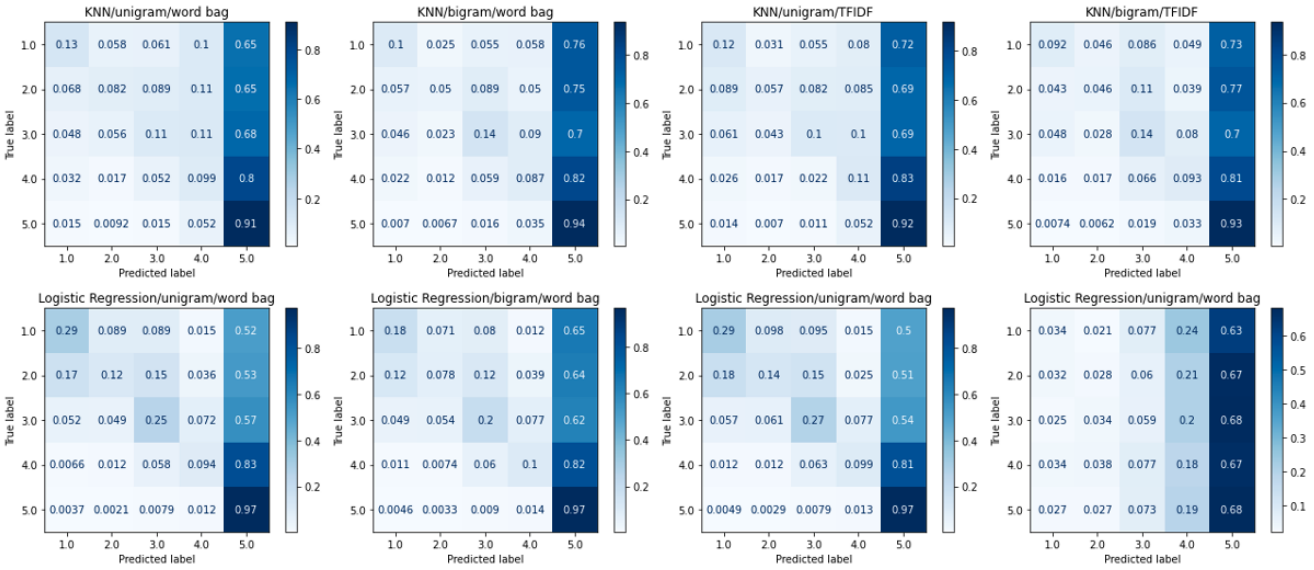

Fig. 6. Contusion matrix of KNN and Logistic Regression

Scheme	Word bag		TFIDF	
Model	unigram	bigram	unigram	bigram
Train	0.734	0.739	0.721	0.733
Validation	0.7264	0.7302	0.724	0.7302
Test	0.7138	0.7013	0.72	0.6788

Table 1. The Result of KNN

Scheme	Word bag		TFIDF	
Model	unigram	bigram	unigram	bigram
Train	0.79002	0.78184	0.79342	0.78114
Validation	0.7776	0.766	0.7767	0.7663
Test	0.7773	0.7667	0.7775	0.7681

Table 2. The Result of Logistic Regression

6.2. Logistic regression

The accuracy of each scheme using Logistic Regression is shown in Table2. In the result of this model, we found that the accuracy of bigram scheme is lower than unigram scheme no matter the input features are word bag or TF-IDF score. It is interesting because word order seems to cause less accuracy when we do prediction. Overall, the accuracy of Logistic Regression is better than KNN. Moreover, we can see that Logistic Regression perform better in low rating prediction through confusion matrix in Figure6.

The problem we suffered during training is that it kept showing warning telling that the model is not convergent. We tried to normalized the data first before training and increase the iteration number. This did solve convergent problems for the model.

7. SUMMARY

In the data, I can know that most customers tend to give five points to the products. In the result, we can see that Logistic Regression perform better than KNN about 8% no matter compared with accuracy or confusion matrix. It proves that KNN model does not work well in high dimensional data and Logistic Regression is a better classifier for my data.

8. REFERENCES

[1] Harris, Zellig (1954). "Distributional Structure".
[2] Minyong Shi, Wenqian Shang, and Zhiguo Hong "Improved Feature Weight Algorithm and Its Application to Text Classification"
[3] Akiko Aizawa "An information-theoretic perspective of tf–idf measures"
[4] Muhammad Zubair Asghar, Aurangzeb Khan, Shakeel Ahmad, Fazal Masud Kundi "A Review of Feature Extraction in Sentiment Analysis"