Sentiment Analysis of Amazon Fine Food Online Reviews (Dataset)

Advanced Database Management & Data Mining (MIS 612)

Phoevi Anngelli E. Lumbao
Master in Information Systems
August 31, 2022

Overview

Nowadays, digital marketers are still obsessed with metrics but often, they forget to consider the buyers' emotions which stand as the first factor in making their buying decisions. Buyers share their feelings and express their thoughts online on any online shopping platform. Like Amazon, since they also sell fine food products online, they need to observe the chatty customers and recognize how these people feel towards the products/brands they carry. To know how buyers feel about Amazon's fine foods based on the reviews and scores, a naïve Bayes sentiment analysis was used. This sentiment analysis will fetch information about a buyers' experience of a product from Amazon; the key elements to helping Amazon understand its market. Sentiment Analysis (or opinion mining) with Python is the chosen methodology for analyzing the Fine Food Online Reviews from Amazon and the platform that was used is the Jupyter notebook.

The main goal is to test if conducting an opinion mining based on food reviews from Amazon as a chosen dataset can be done and worth to know based on the accuracy and prediction reports and with classification whether it's negative or positive sentiments.

The raw dataset available at Kaggle.com contains online reviews of fine foods from Amazon. This was collected for more than a decade beginning from October 1999 - October 2012 with a total of ~500,000 reviews. The reviews include user information and their scores where 1 is the lowest and 5 is the highest, product ID, and a plain text summary and review. After splitting the raw data downloaded, the majority falls under positive sentiment thus, needing to remove much of it for train and test purposes and to meet a balanced sample data; from ~500,000 reviews down to 249,348 only. The following number of samples were used respectively: 90% train dataset or 224,414 data and 10% test dataset or 24,934 data.

In the end, it was determined that the accuracy score of the test and train result of sentiment analysis using the naïve Bayes algorithm is 75%.

General Objective

To test if it is possible to conduct an opinion mining through posted experiences of different people who participated in the reviews for Amazon Fine Food products and classified them as either negative or positive sentiment.

General Procedure of Train and Test of Datasets

1. Download and install Anaconda python and launch the Jupyter notebook.
2. Install all necessary Python Packages to be used in the notebook.

3. Download the dataset from Kaggle.com.
    a. The downloaded file name is Reviews.csv.
4. Create a new file and save it as "TrainDataReview.csv". This new file will be used as a dataset for the train.
    a. There are eleven (11) columns. Add one (1) more column for "sentiment" as this will be used as a reference to the train and test later in the notebook.
5. Assign each sample data based on its content whether it's negative (0) or positive (1) sentiment.
    a. The basis for the assignment is from the Scores. All scores from 1-3 are to be assigned to negative samples; 4-5 scores are to be assigned to positive samples.
6. The most numbered sample is in favor of positive sentiments; thus, there's a need to remove excessive positive samples to have balanced data for the train and test.
7. Save all the changes made in the "TrainDataReview.csv" file**.**
8. Create another file and save it as "TestDataReview.csv". This newly created file will be used as a dataset for the test.
    a. For the test, the total number of sample data is **24,934**, or 10% of the "TrainDataReview.csv" dataset used for the training having a total number of **224,414**.
9. Now that the datasets are ready, go to the Jupyter notebook. *(Refer to the codes with file name: AmazonFineFoodReviews_Sentiment Analysis.ipynb)*
10. Run the **functions**. Wait for the result. It can detect a natural language [nltk_data].
11. Run the **pandas** pd.read_csv() to pull the datasets for the train and test.
12. Run the seaborn sns.countplot(). Wait for the output showing the Sentiment Distribution Result with classification of Negative and Positive.
13. Run the **wordCloud** wordcloud(). Wait for the result. The largest font word that pops-up means the most frequent words occurred during the train while the small font words are the slightly used words that occurred.
14. Run class Tokenizer for the stop words in English.
15. Run class MultinomialNaiveBayes.
    a. Assign the test and train data.
16. Run MultinomialNaiveBayes () for analysis.
17. Run y_hat = MNB.predict(X_test) using MultinomialNaiveBayes to predict the data.
18. Run accuracy_score()
    a. Run print(classification_report(y_test, y_hat)): output comes with the following result: Precision, Recall, F1-score , and support.
19. Run confusion_matrix(). The result reflects the total of predicted and unpredicted data.
20. Run heatmap sns.heatmap() for the final output of the train and test performed.

Opinion Mining Methods

1. Prepare the datasets to be used for opinion mining.
2. Use Jupyter Notebook as a computing platform.
3. Choose the terms for sentiment analysis: negative (0) and positive (1) as sample data.
4. Split the sample data equally.
5. Run the codes in the Jupyter Notebook.
6. Interpret the train and test results.

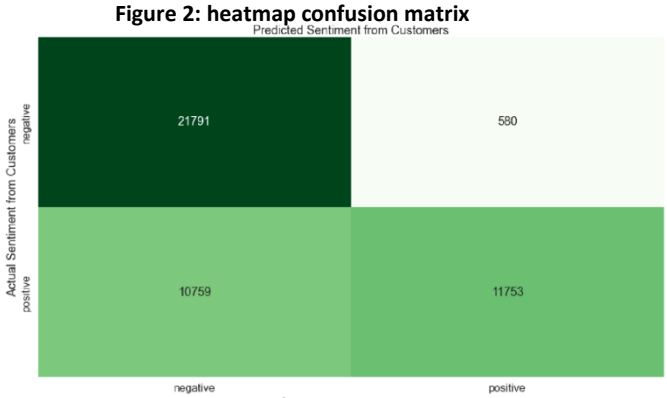
Results

The pulled-out data used for the train was presented correctly as shown in Sentiment Distribution Result where both negative (112,207 samples) and positive (112,207 samples) reviews were properly classified.

Figure 1: Predicting the sample data

	precision	recall	f1-score	support
0	0.67	0.97	0.79	22371
1	0.95	0.52	0.67	22512
accuracy			0.75	44883
macro avg	0.81	0.75	0.73	44883
weighted avg	0.81	0.75	0.73	44883

Figure 2: heatmap confusion matrix



Moreover, based on the Figure 1, it is noted that the accuracy score obtained after the test is **75%** and the result of the predicted vs actual sentiments matrix is **21,791 and 580**; **10,759 and 11,753** *(Refer to Figure 2).*

Discussions

This paper shows that the naïve Bayes algorithm used to correctly classify the sample data reached a 75% accuracy score, a great model performance, and is realistic (Barkved, 2022). To support the accuracy score presented, the confusion matrix was performed. Of the 22,371 negative sample data, there were 21,791 negatives correctly predicted and the other 580 were positive sample data. On the other hand, the algorithm correctly predicted only 11,753 of the 22,512 positive sample data, meaning it got a whopping 10,759 sample data wrong. The f1-score shown in Figure 1 can be interpreted as there's a 79% chance that the new online fine food reviews, let's say one (1) user with one or more reviews in English posted on Amazon will be classified as negative (0), and 67% chance as positive (1). The reason for some reviews may be unpredicted (false negative or false positive) is probably because it's neutral or the words used in reviews are non-English.

By performing naive Bayes algorithm sentiment analysis, interpretation of the results could make it more informative for Amazon to understand the customers' collective experiences posted online. Moreover, knowing this trend will help Amazon understand deeper about their market and improve the products/brands that they carry to make their target market buy more from them.

With the results mentioned, it's indeed possible to conduct opinion mining with a thousand datasets and may be used as future references by Amazon. The result of the chosen algorithm may give you ideal and realistic reports.

References:

J. McAuley and J. Leskovec. *Amateurs To Connoisseurs: Modeling The Evolution Of User Expertise Through Online Reviews.* 2013. Retrieved from Link: http://i.stanford.edu/~julian/pdfs/www13.pdf.

Barkved, K. *How To Know if Your Machine Learning Model Has Good Performance*. 2022. Retrieved from link: https://www.obviously.ai/post/machine-learning-model-performance.