Amazon: analysis on reviews of the 'Grocery and Gourmet Food' category.

Laura Nembrini 819059, Gabriele Strano 866563, Sébastien Marro 882093
Università degli Studi di Milano-Bicocca, CdLM Data Science

13 Gennaio 2022

Abstract

Technological progress has led to a great deal of importance being placed on user-generated content. In fact, nowadays, when purchasing a product or service we tend to rely on the experience of users who have previously used it. In this project, we focus on reviews found on one of the world's largest e-commerce platforms, Amazon. Every day, hundreds of thousands of customers give reviews on the products and services they buy. We particularly focused on reviews belonging to the product category 'grocery and gourmet food' going to apply mainly 2 tasks: classification task that allows, through binary classifier, to predict whether a review provided can be categorized as positive or negative and the task of clustering in order to obtain a breakdown with respect to the polarity of the reviews and be able to make a comparison between the latter and the evaluation associated with users. The first phases were developed around data cleaning and preprocessing of the analyzed data. In order to test the models for classification, we opted for a division of the dataset into training and testing. On the other hand, regarding the second task, sentiment analysis was calculated, useful to extract polarity from the text of the reviews.

Keywords: TextMining, BinaryClassification, SentimentAnalysis, TextClustering, AmazonReviews.

Contents

1 Introduction

In 1994, Jeff Bezos founded Amazon, one of the largest e-commerce platforms in the world. Originally it was an online bookstore but later it extended the type of purchasable products. In fact, now you can buy an endless number of products such as books, CDs, DVDs, video games, beauty products, food, clothes and much more. The birth of this platform has revolutionized the consumption habits of users. In fact, today making a purchase no longer requires the physical presence of consumers, but this activity can be done from the comfort of home. One of the features that has allowed this platform to stand out is the fact that it is customer oriented, meaning that it places emphasis on the customer and their needs. An opportunity that this platform provides is that of being able to review the products purchased. Indeed, after each purchase, there is the possibility to leave a public review, with an associated rating from 1 to 5 stars, which allows you to express your degree of satisfaction. The provided ability to review products allows the customer to interact and rely on the opinions of others to be guided in their purchases.

1.1 Goal

The goal of this project is twofold: on the one hand, it is to build a predictive tool, specifically, a binary classifier that can associate a review with its most appropriate label, which emerges in the form of star ratings; on the other hand, it is to apply the text clustering technique to test whether, in groups of similar reviews, the polarity of the reviews corresponded to the rating provided through stars. The polarity of reviews is calculated through sentiment analysis, which will be described later. Specifically, regarding the binary rating task, it is intended to provide for two classes, the positively rated class (stars assigned to the review may be 4 or 5) and the negatively rated class (stars assigned range from 1 to 3).

2 Dataset

For this project, the Amazon Reviews dataset [1] was used, which is a large collection of user reviews of various products sold on Amazon. In addition to reviews (rating, review text), the dataset includes data referring to the product sold (such as images, price, brand, etc.). The full dataset includes 233.1 million reviews, but for our purpose we chose to use a subset related to 'grocery and gourmet food'. There are 1,143,470 reviews in this dataset.
The variables belonging to the dataset are:

1. overall: rating (in stars) of the product;
2. verified;
3. reviewTime: time of the review;
4. reviewerID: ID of the reviewer;
5. asin: ID of the product;
6. reviewerName: name of the reviewer;
7. reviewText: text of the review;
8. summary: summary of the review;
9. unixReviewTime: time of the review (unix time);
10. vote: useful vote of the review;
11. style: a dictionary of the product metadata;
12. image: images that users post after receiving the product.

The features of interest for this project are 'reviewText', i.e., the textual corpus related to the reviews and 'overall' i.e., the score associated to the review, which correspond to the number of stars with which you evaluate the purchased product. The score varies from 1 to 5 stars (corresponding respectively to a low and a high satisfaction). As a preliminary operation, all those reviews with a null value were removed. This allowed to reduce, even if slightly, the dimensionality of the dataset. It was then possible to visualize, through the following plot, an imbalance between the various classes in the overall column.

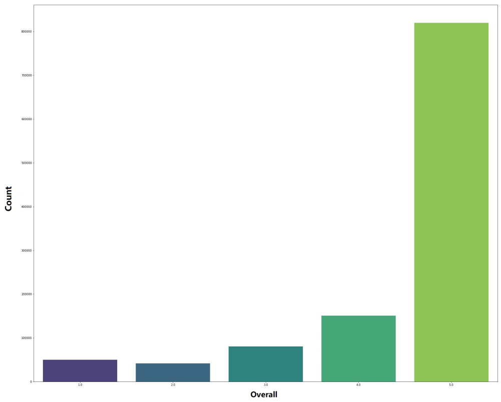

Figure 1: Unbalanced classes barchart.

In fact, it is possible to note, for example, the strong disproportion between the class with two stars, and that with 5 stars. Therefore, we chose to consider mainly two classes: the class with positive polarity, consisting of those reviews with a score greater than 3, and the class with negative polarity, in which there are reviews with a score less than or equal to 3. At this point, in order to obtain a balanced dataset, 126384 instances of positive rating and 126384 of negative rating were considered, divided equally between the scores belonging to the two classes. This means that, for example, in the class with negative polarity, $\frac{1}{3}$ of the 126384 reviews with a 1-star rating, $\frac{1}{3}$ with a 2-star rating, and $\frac{1}{3}$ with a 3-star rating will be considered.

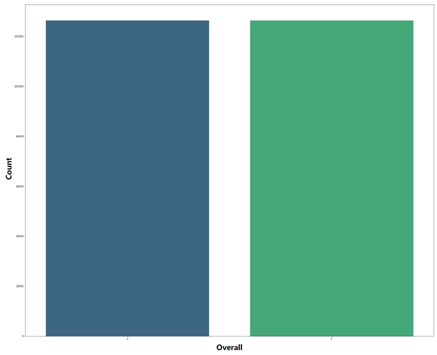

Figure 2: Balanced classes barchart.

3 Methodological approach

3.1 Pre-processing

Text preprocessing is a key step in NLP, or Natural Language Processing, activities. These preprocessing techniques allow the text to be transformed into a more digestible form so that Machine Learning algorithms can perform better. At a preliminary stage, an exploratory analysis was performed in order to highlight the most used words within the entire corpus of reviews, as can be seen from the following wordcloud.

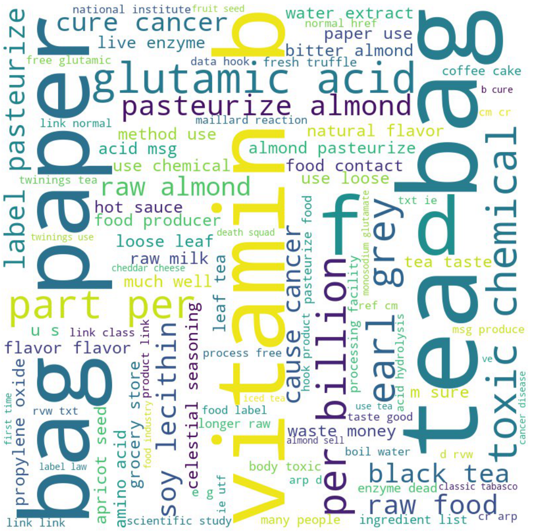

Figure 3: Wordcloud representing the highest frequency words.

As a first step, a cleaning of the text of the reviews was carried out. Specifically, it was applied:

1. *Lower case*: allows you to make the words uniform from the point of view of the font format. Converts all text to lowercase so that the computer does not treat two identical words as different just because they have one letter written in different versions.

2. *Removing special characters and punctuation*: removal emoticons and other characters, as they are not useful to understand the degree of satisfaction of the review.

After homogenizing the text and clearing it of superfluous characters, we moved on to the actual pre-processing phase. The following is a detailed explanation of the various operations considered:

1. *Tokenization*: allows the text to be broken down into smaller units. This step is necessary to split each review that is stored as a string into multiple tokens consisting of individual words.

2. *Stop words*: all the typical words of the language that are redundant and therefore not useful for extracting the meaning of the text are removed. They are part of the stopwords: articles, prepositions, conjunctions. The purpose is to reduce the length of the review while maintaining its informativeness.

3. *Lemmatization and POS*: lemmatization allows words to be traced back to their basic form. In this way, singular or plural forms of the same word are replaced with the same term. The *NLTK* library was also used for this step. To make lemmatization more effective, Part Of Speech tagging was carried out, which allows categorizing the words of a text at a particular part of speech. POS helps lemmatization to avoid errors on the lexical category and thus allows you to avoid transforming words that do not need to be transformed.

3.2 Text Representation

Text representation is one of the fundamental problems in the world of Text Mining and Information Retrieval. It aims at numerically representing unstructured text documents in such a way that they are suitable for the work of algorithms. As a textual representation, we chose to use TF-IDF, a function that allows us to measure the importance of a term with respect to a document or collection of documents. It considers the information related to the frequency with which terms appear, which is normalized with respect to the length of the reviews. It also allows us to give less weight to the most common words in the corpus and which have a low discriminatory power. In fact, the value of the function increases proportionally to the number of times the term is contained in the document but increases inversely to the frequency of the term in the whole corpus.
Given the sparsity and size of the matrix obtained, a dimensionality reduction technique was applied, in this case the truncated Singular Value Decomposition (SVD) technique was applied. The number of components has been set to 100 (recommended value for our purpose) [2]. Although these new features are not easily interpretable, they are excellent for mathematical modeling. The result is in fact a matrix with a smaller size than the previous one, which therefore allows to reduce the computational weight and whose percentage of variance explained is equal to 95%, thus allowing to preserve most of the information.

3.3 Text Classification

As mentioned earlier, the first goal of the project is to implement a binary classification, i.e., predictive tool that has the purpose of associating a review to its most appropriate label, in this case 0 if it is a negative review and 1 if it is a positive review. Before tackling the actual implementation phase, a split threshold between train and test was established at 85% and 15%, respectively. For this purpose, it was chosen to use a neural network for classification. A recurrent neural network of type LSTM was implemented using the *keras* library. Below are the specifications of the network:

- *Embedding layer*: vocabulary size is 100000 and vector length is 100;

- *LSTM layer*: consists of 128 neurons. The parameters 'dropout' and 'recurrent_dropout' equal to 0.5 were inserted in order to limit overfitting;

- *Output layer*: dense layer consisting of 2 neurons, one per category, with sigmoid activation function.

The adam optimizer, the categorical crossentropy loss function, was chosen for network fitting. Once the specifications were defined, the network was trained using 4 epochs and a batch size of 1024.

3.4 Text Clustering

The second goal posed was to get a breakdown of the reviews with respect to the polarity of the reviews and to be able to make a comparison between the latter and the rating associated by the users. For the realization of the unsupervised clustering, the K-Means algorithm (served by the *sklearn* library) was chosen. The clustering is performed on the polarities of the reviews that is extracted through Sentiment Analysis, calculated through the *NLTK* library and the users' evaluation. From this feature, we derived a dictionary containing 4 keys: positive, negative, neutral, and compound, followed by their respective values. For the clustering task, after several attempts we realized that the value that allowed more than the others to discriminate the various groups was the compound value. The compound score is computed by summing the valence scores of each word in the lexicon, adjusted according to the rules, and then normalized to be between -1 (most extreme negative) and +1 (most extreme positive). These polarities are then passed to the K-Means algorithm, which iteratively computes the centroids using Euclidean distance. The number of clusters is set equal to 4.

4 Results

This section shows the results obtained from the two applied tasks.

4.1 Text Classification

The results obtained are shown in the table below:

	Precision	Recall	f1-score	support
negative	0.85	0.84	0.84	19146
positive	0.84	0.84	0.84	18770
accuracy			0.84	37916
macro avg	0.84	0.84	0.84	37916
weighted avg	0.84	0.84	0.84	37916

Table 1: Text classification results.

The predictor used has excellent results for both classes, this has mainly confirmed the goodness of the choice to work with well balanced classes even at the cost of reducing the dimensionality of the documents available. Even the precision values are good so it was possible to have a good reliability on the predicted class without making the algorithm too selective, in fact even the recall values are quite similar. The results obtained can also be improved by increasing the training epochs and the input data, but this would have meant less efficiency from a computational point of view.

4.2 Text Clustering

The following graph shows the results:

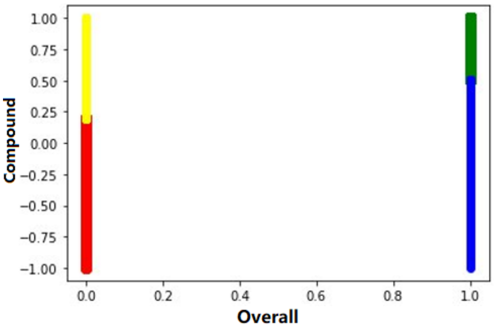

Figure 4: Text clustering results.

Among the 4 clusters formed (clusters from 0 to 3), those of greatest interest for our analysis were cluster number 0 (red cluster in the graph) and cluster number 1 (blue cluster in the graph), which were those for which the two metrics considered, that is, the sentiment analysis score and the overall dichotomized score, were in agreement. These clusters, in our interpretation, are therefore the most reliable for further analysis, since with this double approach the conformity of the star rating with what was written in the reviews was ascertained.

5 Conclusion and future developements

Through the pre-processing phase and the two subsequent tasks, it was possible to analyze the content of the textual reviews. The text classification approach can be very useful when there is a large amount of reviews to be analyzed, especially in contexts in which, unlike the one we are dealing with, we do not have a column (overall) summarizing the degree of satisfaction. The text clustering phase, on the other hand, is useful in order to extract hidden information from the structure of the data; in the case treated, 2 clusters of reviews emerged which present greater reliability with regard to the classification received.

Future developments include the possibility of considering parts of the text that could make a significant contribution to understanding the review provided. This is the case of emoji, which, in the field of Sentiment Analysis and Irony Detection, could help to better understand the meaning of the text. In fact, especially in negative reviews, irony is often present and being able to catch it could be an additional value, even if it is challenging. Moreover, you can then go and consider all the reviews on the Amazon platform and not just those related to the 'grocery and gourmet food category', so not only would the analysis be extended to more products, but the prediction algorithm used in the text classification task would have much more "material" to improve during the learning phase.

References

[1] https://nijianmo.github.io/amazon/index.html.

[2] https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.TruncatedSVD.html.