Sentiment Classification of Amazon Reviews based on Naives Bayes and Logistic Regression, and Dimensionality Reduction Methods

Jorge Armando Barrera Ceballos
College of Computer, Mathematical, &
Natural Sciences
University of Maryland
College Park
jbarrer2@umd.edu

***Abstract*—An analysis of the classification of Amazon reviews into positive and negative was carried out, and the features with the greatest capacity for classification of positive and negative reviews were identified. It is relevant because those factors that positively or negatively impact customer evaluation could be identified. For this, Naives Bayes and Logistics Regression classification methods were used, with feature selection methods such as SelecKbest, Truncated SVD, and n-grams. Prior to the application of the models and the dimensionality reduction, the texts of the Amazon reviews were transformed through Translation, Lowerization, Tokenization, removal of Stop words, and Lemmatization. The evaluation metrics used are Accuracy, Precision, Recall, and the Receiver Operating Characteristic Area Under the Curve (ROC AUC). As a result, levels close to 0.9 were obtained for the Accuracy, Precision, and Recall metrics, and a value close to 0.8 for the ROC AUC metric. It was possible to identify the features with the greatest power to classify positive feelings: great, perfect, love, excellent, among others. Also, it was possible to identify the features with the greatest power to classify negative feelings: disappointing, useless, and returning.**

Keywords—classification, transformations, dimensionality, reduction, metrics, positive, negative, reviews.

I. Introduction

This article aims to classify Amazon reviews into positive and negative based on the text that customers write as a review after making their purchase or even after receiving their merchandise. The importance of this analysis lies in the fact that those factors that lead a customer to give a positive or negative review could be identified and addressed later. These factors could include the quality of the product, delivery time, usefulness, packaging, whether it met expectations, among many others. For this, it is first necessary to have a model with a high capacity to discriminate between positive and negative reviews.

The dataset consists of Amazon reviews from the Musical Instruments category, which contains more than 1.5 million reviews, and is available on the University of California, San Diego website.

For this purpose, the Naives Bayes and Logistic Regression classification methods were used. To generate the models, the texts of the Amazon reviews were first processed accordingly using Translation, Lowerization, Tokenization, removal of Stop words, and Lemmatization. The resulting tokens were represented in a matrix using the Term Frequency and Inverse Document Frequency TF-IDF scores, to introduce them as features in the models.

Because the number of features is very large, greater than 120,000, dimension reduction methods such as SelecKbest, Truncated SVD, and n-grams were used. To evaluate the models, performance metrics were used: Accuracy, Precision, Recall, and the Receiver Operating Characteristic Area Under the Curve (ROC AUC).

From the Naive Bayes and Logistic Regression models used, performance metrics greater than 0.8 were obtained. The best model was the Logistic Regression model with dimension reduction to 1,000 features, with which values of the evaluation metrics close to 0.9 were obtained.

It is also important to highlight that the most important positive and negative features were identified using the estimated coefficients of the Logistic Regression model. The features with the greatest power to classify positive reviews are great, perfect, love excellent, among others., and the negative features with the greatest capacity to discriminate reviews are disappointing, useless, and returning.

II. Background

In a review of the literature on classification of reviews to extract sentiments, in [1] a sentiment analysis of reviews was carried out using Naive Bayes on laptop products, in which accuracy levels close to 0.9 were obtained for the best model, as in this article. However, in that article the best classification results were obtained using bigrams.

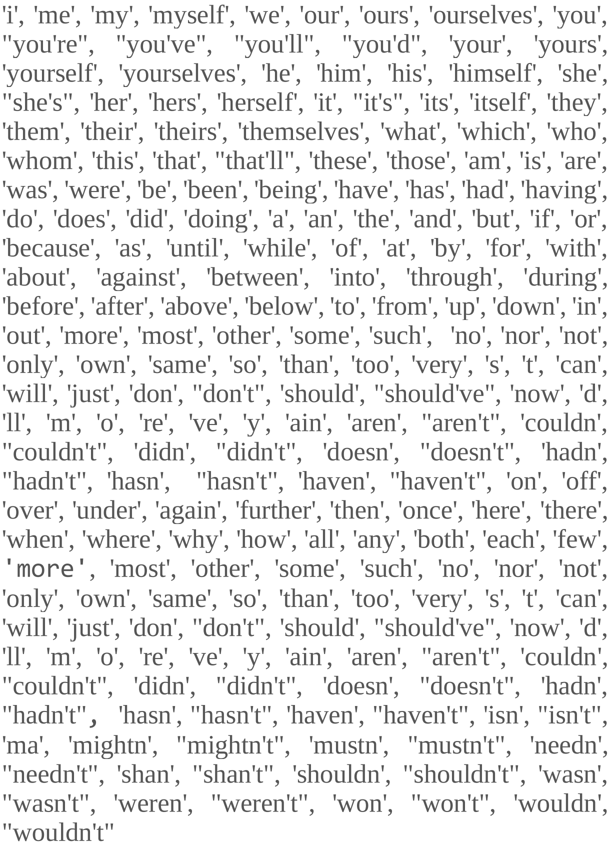
In [2] other additional methods to Naives Bayes and Logistic Regression were used such as Random Forests, Bidirectional Long-Short Term Memory (LSTM), and Bert. In addition to TF-IDF, Bag-of-Words and GloVE were considered as feature extraction methods. Performance metrics such as accuracy, precision and recall were also used. The values obtained for the Naives Bayes and Logistic Regression models were lower than those obtained in this article. However, the values obtained with the LSTM and BERT models were higher, ranging between 0.93 and 0.96. The best performance metrics were obtained with the BERT model; however, no feature extraction method was used for this model.

III. Methodology

A. Data Description

A database of Amazon reviews from the Musical Instruments category was analyzed, which contains more than 1,500,000 reviews. The relevant dataset variables for this work are the full text of each review and a rating between 1 and 5 given by the client for each review.

Other variables that make up the database are the name of the person who made the review and a summary of the review. The data was obtained from the Amazon Review Data (2018) website of the University of California at San Diego [3].



'i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you're", "you've", "you'll", "you'd", 'your', 'yours', 'yourself', 'yourselves', 'he', 'him', 'his', 'himself', 'she', "she's", 'her', 'hers', 'herself', 'it', "it's", 'its', 'itself', 'they', 'them', 'their', 'theirs', 'themselves', 'what', 'which', 'who', 'whom', 'this', 'that', "that'll", 'these', 'those', 'am', 'is', 'are', 'was', 'were', 'be', 'been', 'being', 'have', 'has', 'had', 'having', 'do', 'does', 'did', 'doing', 'a', 'an', 'the', 'and', 'but', 'if', 'or', 'because', 'as', 'until', 'while', 'of', 'at', 'by', 'for', 'with', 'about', 'against', 'between', 'into', 'through', 'during', 'before', 'after', 'above', 'below', 'to', 'from', 'up', 'down', 'in', 'out', 'more', 'most', 'other', 'some', 'such', 'no', 'nor', 'not', 'only', 'own', 'same', 'so', 'than', 'too', 'very', 's', 't', 'can', 'will', 'just', 'don', "don't", 'should', "should've", 'now', 'd', 'll', 'm', 'o', 're', 've', 'y', 'ain', 'aren', "aren't", 'couldn', "couldn't", 'didn', "didn't", 'doesn', "doesn't", 'hadn', "hadn't", 'hasn', "hasn't", 'haven', "haven't", 'on', 'off', 'over', 'under', 'again', 'further', 'then', 'once', 'here', 'there', 'when', 'where', 'why', 'how', 'all', 'any', 'both', 'each', 'few', 'more', 'most', 'other', 'some', 'such', 'no', 'nor', 'not', 'only', 'own', 'same', 'so', 'than', 'too', 'very', 's', 't', 'can', 'will', 'just', 'don', "don't", 'should', "should've", 'now', 'd', 'll', 'm', 'o', 're', 've', 'y', 'ain', 'aren', "aren't", 'couldn', "couldn't", 'didn', "didn't", 'doesn', "doesn't", 'hadn', "hadn't", 'hasn', "hasn't", 'haven', "haven't", 'isn', "isn't", 'ma', 'mightn', "mightn't", 'mustn', "mustn't", 'needn', "needn't", 'shan', "shan't", 'shouldn', "shouldn't", 'wasn', "wasn't", 'weren', "weren't", 'won', "won't", 'wouldn', "wouldn't"

Fig 1. *NLTK stop words.* (*It can be seen that negative stop words such as "aren't", "couldn't", "didn't", "doesn't", "hadn't", etc., and other variations of these words are removed. This is mentioned because the performance metrics of the reviews could be biased, resulting in higher values of the metrics for positive reviews).*

B. Data Preprocessing

To prepare the data for the analysis, records with missing data in the review text or in the rating were eliminated, bringing the final number of reviews to 1,511,172. As the next step, a dichotomous variable was created from the variable that contains the rating. The number of cases in each category is presented in Table I. The categories of the dichotomous variable are unbalanced, which will be mentioned and treated in the "Methods" section.

TABLE I. CODING OF THE RATING VARIABLE

Creation of the label variable in the data set		
Overall	***Label***	***Relative Number of Reviews***
1 to 3	0 (negative)	19.3%
4 and 5	1 (positive)	80.7%

To work with the review text variable, the following transformations were carried out prior to analysis: i) Translation, punctuation marks were removed from the reviews; ii) Lowerization, all letters were converted to lowercase; iii) Tokenization, each review was divided into substrings; iv) Stop words, the words considered stop in the English language were eliminated [4]; v) Lemmatization, the lemma of a word was determined based on its intended meaning and its context. E.g., "better" has "Good" as its motto, and "walk" is the base of "walking" [5].

Fig. 1 shows the set of NLTK stop words used. With this set negative stop words such as "aren't", "couldn't", "didn't", "doesn't", "hadn't", etc., and other variations of these are removed. This is mentioned because the performance metrics of the reviews could be biased, resulting in higher values of the metrics for positive reviews.

The reviews were then converted into a matrix of TF-IDF features, where TF stands for Term Frequency and IDF stands for Inverse Document Frequency. In mathematical terms [4]:

$$tf_{ij} = frequency\ of\ word\ j\ in\ document\ i$$

$$idf_j = \log\left(\#documents\ /\ \#documents\ with\ word\ j\ \right)$$

$$TF - IDF \text{ score} = \ tf_{ij}\ x\ idf_j$$

The *TF* is generally overloaded with very frequent words, so it is negatively weighted by the term *IDF*, which reflects how frequently a word appears in a corpus or a collection of texts.

To exemplify the calculation of the *TF-IDF* indicator, the following set of texts is analyzed below:

['This is the first document', 'This document is the second document', 'And this is the third one']

The first step is to create the lexical:

['and', 'document', 'first', 'is', 'one', 'second', 'the', 'third', 'this']

Next, the matrix representation of the document of the set of texts is calculated, which has dimension 4x9, and which appears in Fig. 2. The word 'and' appears 0 times in the first text, which is represented in the matrix with a value of 0 in the first entry of the matrix.

$$\begin{bmatrix} 0\,1\,1\,1\,0\,0\,1\,0\,1 \\ 0\,2\,0\,1\,0\,1\,1\,0\,1 \\ 1\,0\,0\,1\,1\,0\,1\,1\,1 \end{bmatrix}$$

Fig 2. Matrix representation of the example texts. (*The word document appears twice in the second text 'This document is the second document').*

Fig. 3 shows the scores that result from the *TF-IDF* transformation. The word 'document' is the most relevant in this example.

$$\begin{bmatrix} 0 & 0.5 & 0.6 & 0.4 & 0 & 0 & 0.4 & 0 & 0.38 \\ 0 & 0.7 & 0 & 0.3 & 0 & 0.5 & 0.3 & 0 & 0.28 \\ 0.5 & 0 & 0 & 0.3 & 0.5 & 0 & 0.3 & 0.5 & 0.27 \end{bmatrix}$$

Fig 3. *TF-IDF* scores. (*A weight has been assigned according to the relevance of each word, such as the word 'document', which is the most relevant and for this reason it obtained the highest score.).*

C. Feature Selection

The following available methods were considered: i) *SelecKbest*, is based on univariate selection of features from univariate statistical tests [6]; ii) *Truncated SVD*, dimensionality reduction is performed based on truncated singular value decomposition that works efficiently with sparse matrices [7]; iii) *n-grams*, the probability of the last word of a subsequence of n words is estimated, given the previous words [8, 9].

Regarding the *Truncated SVD* method, when applied to a data matrix of training data X:

$$X \approx X_k = U_k \Sigma_k V_k$$

After this operation, the transformed training set with k features is $U_k \Sigma_k$. To also transform the test set, the following operation is performed:

$$X' = XV_k$$

D. Classification Methods

The following methods were considered: i) *Naive Bayes*, based on the Bayes' theorem under the “naive” assumption of conditional independence between each pair of features given the value of the class variable; ii) *Logistic Regression*, which is a special case of a Generalized Linear Model with a Binomial / Bernoulli conditional distribution and a Logit link. The numerical output is used as the estimated probability for classification, given a certain threshold [10].

The Bayes' theorem establishes the following relationship between the target variable that takes values in the set {0,1}, and the feature vector that contains the features $x_1, x_2, \dots, x_n$ [11]:

$$P(y\,|x_1, x_2, \dots, x_n) = \frac{P(y)P(x_1, x_2, \dots, x_n|y)}{P(x_1, x_2, \dots, x_n)}$$

By using the assumption of independence, we obtain:

$$P(\,x_i\,|y, x_1, x_2, \dots, x_{i-1}, x_{i+1}, \dots, x_n) = P(x_i|y)$$

for all *i*, which can be simplified as:

$$P(y\,|x_1, x_2, \dots, x_n) = \frac{P(y)\prod_{i=1}^{n} P(x_i|y)}{P(x_1, x_2, \dots, x_n)}$$

Because in this case $P(x_1, x_2, \dots, x_n)$ is a constant, it is possible to use the following classification rule:

$$P(y\,|x_1, x_2, \dots, x_n)\ \alpha\ P(y)\ \prod_{i=1}^{n} P(x_i|y)$$

Which implies that:

$$\hat{y} = \arg\max_y P(y)\ \prod_{i=1}^{n} P(x_i|y)$$

Regarding Logistic Regression, this method estimates the probabilities that each review belongs to one of the two classes, 0 or 1, by minimizing the following cost function [12]:

$$C(x_1, x_2, \dots, x_n, w) =$$
$$-y_i \ln\big(g(x_1, x_2, \dots, x_n)\big) - (1 - y_i) \ln\big(g(x_1, x_2, \dots, x_n)\big)$$

Where:

$$g(x_1, x_2, \dots, x_n) = \frac{1}{1 + e^{-w^T x}}$$

and w is a vector of coefficients w_i for each feature x_i.

E. Training

The data was separated into training data (70%) and testing data (30%). To prevent overfitting, k-fold cross-validation with k = 5 was used. Because the classes of the dichotomous variable are unbalanced, stratification was used in the cross-validation. A visual explanation of the cross-validation method is presented in Fig. 4.

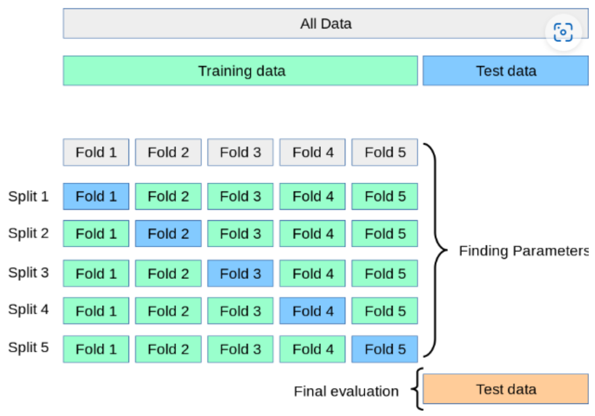


Fig 4. *k*-fold cross validation. (*It can be seen that different splits of the training data are made, in this case 5. In each split, k – 1 of the folds are used as training data, and the remaining part is used as a test set to compute a performance metric such as Accuracy. The performance metric reports are the average of the computed values in each split. A final evaluation is carried out in the test set).*

F. Performance Metrics

The following evaluation metrics were used: i) Accuracy, is the percentage of cases correctly classified; ii) Precision, the ability of the classifier to not label as positive a sample that it is negative; iii) Recall, the ability of the classifier to find all the positive samples; iv) Receiver Operating Characteristic Area Under the Curve (ROC AUC), that is the overall performance of the classifier. The following equations define the metrics used [2]:

$$Accuracy = \frac{tp + tn}{tp + tn + fp + fn}$$

$$Precision = \frac{tp}{tp + fp}$$

$$Recall = \frac{tp}{tp + fn}$$

Where:

$$tp = true\ positives$$

$$tn = true\ negative$$

$$fp = false\ positive$$

$$fn = false\ negative$$

The ROC curve is used in binary classification problems like the one discussed in this article. It plots *(1 – Specificity)* versus Recall, where Specificity is the proportion of true negative cases:

$$Specificity = \frac{tn}{tn + fp}$$

G. Software

The data readiness and data analysis were carried out in Databricks.

IV. Results

Higher performance metrics were obtained with Logistic Regression models than with Naives Bayes models. Furthermore, the *SelecKbest* method with *k = 1000* was the best for selecting features with the greatest discriminatory power.

In Table II. the different adjusted models and each of the evaluation metrics are presented. For the Naive Bayes method, higher levels of classification were obtained with 2-grams. In contrast, with the Logistic Regression method, higher values of the metrics were obtained if only the *SelecKbest* method with *k = 1000* was used.

TABLE II. MODEL EVALUATION

Model	**Evaluation Metric**				
	Feature Selection	***Accuracy***	***Precision***	***Recall***	***ROC AUC***
Naïve Bayes	*SelecKbest (k = 1000)*	0.83	0.82	0.83	0.70
	SelecKbest (k = 1000) +Trucanted SVD (n = 200)	0.79	0.82	0.79	0.72
	SelecKbest (k = 1000) +2-gram	0.86	0.85	0.86	0.68
	SelecKbest (k = 1000) +3-gram	0.83	0.82	0.83	0.59
Logistic Regression	*SelecKbest (k = 1000)*	0.89	0.88	0.89	0.77
	SelecKbest (k = 1000) +Trucanted SVD (n = 200)	0.87	0.86	0.87	0.72
	SelecKbest (k = 1000) +2-gram	0.85	0.85	0.85	0.64
	SelecKbest (k = 1000) +3-gram	0.82	0.83	0.82	0.53

In Table III the execution times for the models considered are presented. In general, the Naives Bayes method is faster than the Logistic Regression method. The model with the best classification results took less than a minute to finish. It is important to mention that these are very short times because the data set has more than 1.5 million reviews, and stratified cross-validation was also carried out for each model.

TABLE III. EXECUTION TIMES

Model	***Feature Selection***	***Execution time***
Naïve Bayes	*SelecKbest (k = 1000)*	30.6 seconds
	SelecKbest (k = 1000) +Trucanted SVD (n = 200)	3.52 minutes
	SelecKbest (k = 1000) +2-gram	1.46 minutes
	SelecKbest (k = 1000) +3-gram	2.94 minutes
Logistic Regression	*SelecKbest (k = 1000)*	58.6 seconds
	SelecKbest (k = 1000) +Trucanted SVD (n = 200)	5.14 minutes
	SelecKbest (k = 1000) +2-gram	1.85 minutes
	SelecKbest (k = 1000) +3-gram	3.24 minutes

Fig. 5 shows the top 20 positive and negative features obtained with the model with the best results. The features with the greatest classification power of positive reviews are great, perfect, love excellent, among others. The negative features with the greatest ability to discriminate reviews are disappointing, useless, and returning.

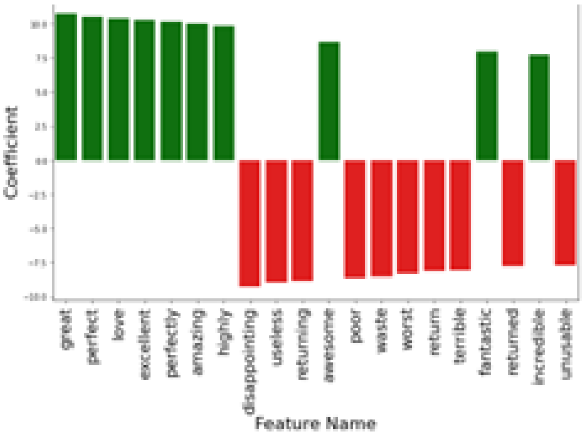


Fig 5. Top 20 features. (*Among the features with the greatest classification power of positive reviews are great, perfect, love excellent, among others. The negative features with the greatest ability to discriminate reviews are disappointing, useless, and returning).*

V. CONCLUSIONS AND NEXT STEPS

Different classification and feature selection methods were used to classify Amazon reviews into positive and negative. The category used was Musical Instruments, which consists of more than 1,500,000 records.

When preparing the data, missing data were eliminated, and the rating variable ranging from 1 to 5 was coded into a dichotomous variable to indicate positive or negative feelings. In addition, the following transformations were performed on the text variable containing the reviews: Translation, Lowerization, Tokenization, removal of Stop words, and Lemmatization.

Among the classification methods explored are Naives Bayes and Logistic Regression. The feature selection methods used are: *SelecKbest*, *Truncated SVD*, and *n-grams.*

Regarding the training phase and to prevent overfitting, the *k*-fold cross-validation method with *k* = 5 was used with stratification, due to the imbalance in class size. The data was separated into training data (70%) and testing data (30%). The evaluation metrics used are Accuracy, Precision, Recall, and the ROC AUC.

The highest values of the evaluation metrics were obtained using the Logistic Regression model, higher than 0.87, except for the ROC AUC in which a value of 0.77 was obtained. These results indicate that by applying traditional classification methods, in conjunction with feature selection, and appropriate data preprocessing methods, it is possible to obtain reasonable values of performance metrics.

It is useful to identify the features with the greatest capacity to classify reviews into positive and negative. Among the features with the greatest capacity to discriminate positive reviews are great, perfect, love excellent, among others. The features with the greatest capacity to discriminate negative reviews are disappointing, useless, and returning.

Two main actions were identified as next steps: i) using other classification methods, such as support vector machines random forests, and Bert; 2) the use of another set of stop words different from that of NLTK. This set of words contains many negative words, which could bias the evaluation metrics towards higher values for positive labels; 3) filter the database by a particular product, which could increase the discrimination capacity and could also provide greater insights; 4) use the categories available in SocialSent for other review datasets. It was not possible to apply it for the Musical Instruments category because the same category does not exist on Reddit. The closest category is Music, however, although it is related, it is not applicable.

REFERENCES

[1] K.H. Mohan, P.S. Sana, and R. Sasikala, “Sentimental analysis of Amazon reviews using naïve bayes on laptop products with MongoDb and R,” IOP Conference Series.Materials Science and Engineering, 263, vol. 4, 2017.

[2] A.S. AlQahtani, “Product sentiment analysis for amazon reviews,” International Journal of Computer Science & Information Technology, vol. 13, June 2021.

[3] Amazon Review Data (2018). “Files – Complete review data,” [Online]. Available: Amazon review data (nijianmo.github.io) [Accessed December 12, 2023].

[4] NLTK Documentation. “Stop words,” [Online]. Available: NLTK: Search [Accessed December 12, 2023].

[5] Wikipedia. “Lemmatization,” [Online]. Available: Lemmatization - Wikipedia [Accessed December 12, 2023].

[6] Scikit-learn. “Feature Selection,” [Online]. Available: 1.13. Feature selection — scikit-learn 1.3.2 documentation [Accessed December 12, 2023].

[7] Scikit-learn. “TruncatedSVD,” [Online]. Available: sklearn.decomposition.TruncatedSVD — scikit-learn 1.3.2 documentation [Accessed December 12, 2023].

[8] Data602 Introduction to Data Science course, University of Maryland, Fall 2022, “Information Extraction and Natural Language Processing”. Not published [Accessed December 12, 2023].

[9] Stanford University. “N-gram Language Models,” [Online]. Available: 3.pdf (stanford.edu) [Accessed December 12, 2023].

[10] Scikit-learn. “Logistic Regression,” [Online]. Available: 1.1. Linear Models — scikit-learn 1.3.2 documentation [Accessed December 12, 2023].

[11] Scikit-learn. “Naïve Bayes,” [Online]. Available: 1.9. Naive Bayes — scikit-learn 1.3.2 documentation [Accessed December 12, 2023].

[12] Data603 Introduction to Machine Learning course, University of Maryland, Fall 2022, “Logistic Regression”. Not published [Accessed December 12, 2023].