Sentiment Analysis for Amazon Product Reviews Using Logistic Regression Model

Journal of Development Economics and MANAGEMENT RESEARCH STUDIES (JDMS) *A PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL*
ISSN: 2582 5119 (ONLINE)
Crossref Prefix No: 10.53422
09(11), 29-42, January-March, 2022
@Center for Development Economic Studies (CDES)
Reprints and permissions
https://www.cdes.org.in/
https://www.cdes.org.in/about-journal/

Sentiment Analysis for Amazon Product Reviews Using Logistic Regression Model

Dr. Prashant Pareek[1], Dr. Neha Sharma[2], Mr. Roushan Raj [3], and Mr. Chetan Dalwadi [4]

Abstract

The sentiment is something that expresses the feeling about anything. This work proposes a Machine Learning approach that performs the classification of customer product reviews by finding the sentiment of the review dataset making it easy for product owners to track and improve their quality of the product rather than going through each piece of ratings and review. Amazon Product Reviews dataset is used for training and predicting systems, aiming to achieve maximum accuracy from the algorithm. This work also uses two different modules like Count Vectorizer and Tf-IDF Vectorizer for comparing the accuracy and consistency in training the text dataset, to conclude the better one between them.

Keywords: Amazon Customer Reviews, Classification, Count Vectorizer, Machine Learning, Tf-IDF Vectorizer

1. Introduction

Sentiment Analysis is one of the NLP (Natural Language Processing) techniques that is used to determine whether data says something positive, negative or neutral about anything. It is performed on textual data and can help businesses monitoring for any brand and maintain their values in the market.

Every message or information can be broadly classified into two main categories such as facts and opinions. Facts are objective statements about entities and worldly events. On the

[1] Assistant Professor, Shanti Business School, Ahmedabad, Gujarat.
[2] Director, Shanti Business School, Ahmedabad, Gujarat, India.
[3] Pursuing MSC in Big Data, FOM University, Germany.
[4] Pursuing MSC in Big Data, FOM University, Germany.

other hand, opinions are subjective statements that reflect people's sentiments or perceptions about the entities and events [3].

Moreover, in today's world, buyers are more intended to check the reviews before buying online products. These product reviews are also significantly utilized to learn the sentiments. On Amazon, the rating can be between 1 to 5. Here, 1 states the worst while 5 denotes the best on a satisfactory level. But, in some instances, there is a mismatch between customers' reviews and ratings. So, it is important to find those mismatched ratings, as they may affect the result of the algorithm [1].

If the user talks about any product, this model gives the resulting sentiment as Positive or negative in the binary form, where 1 denotes the positive and 0 denotes the negative. For example, if a review reads like "This is a very good product...I love the functional features of this...", then the review will be marked as positive as the words "love", "good" etc. are of positive sentiments. Similar things go for negative sentiments and reviews.

2. Work Flow

The model of this system-flow describes the total completion from beginning to end. It describes the sequence in a simple manner. First of all, this system acquires the data which is collected from Kaggle. The dataset is in a zipped format as it contains a lot of reviews. It has separate Training Data and Testing Data with a huge number of reviews.

Then, some pre-processing of the data is done. It contains a lot of technicalities which are discussed in the paper later in detail. After the pre-processing of the data, the LR (Logistic Regression) model is built. Here, we use the dividing of data into a training set and testing set, by split function. The training and testing data ratio which is kept is 3:1, that is like 75% of the data is split and used for training the model, while 25% of data are used for testing the model. Then the model is regularized. On executing the LR model of the system, the model shows the results and it can be seen how accurately the system is giving the result. Classified results of each of the reviews can be seen, which will be given in a binary form, i.e., 1(positive) or 0(negative).

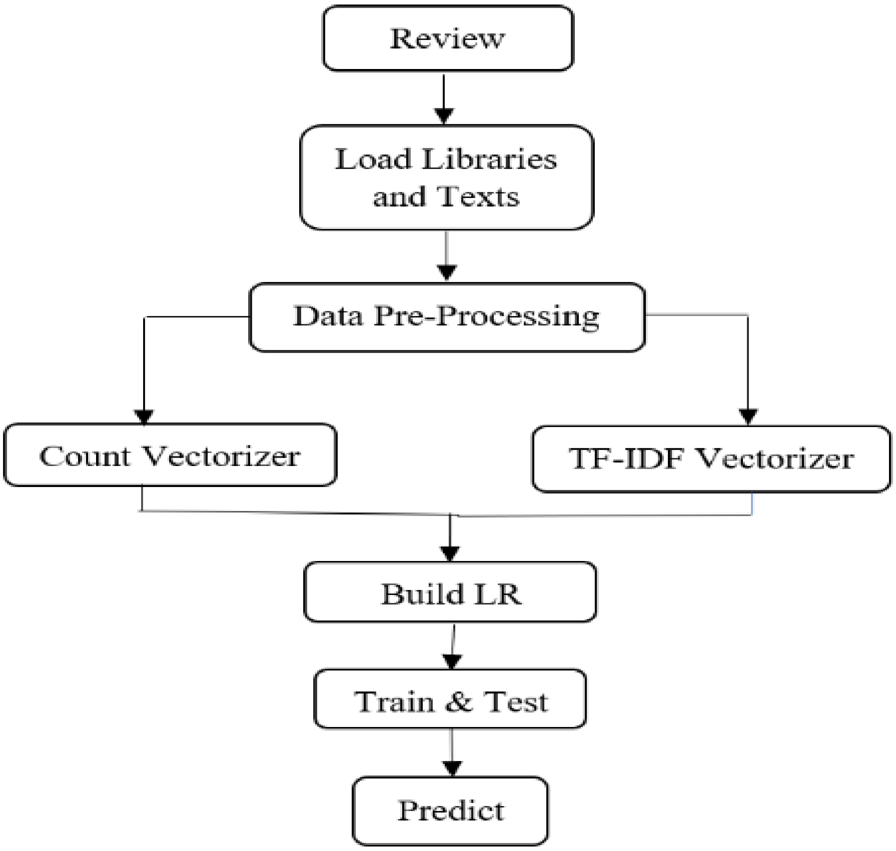


Figure 1: Flow Diagram of the Module

Figure 1 describes the algorithm run right from the beginning of the system to how it is going to predict the output.

3. Methodology

3.1 Data Acquisition

The Dataset which is being used by the system is extracted from Kaggle (Amazon Reviews for Sentiment Analysis). It has some lakhs of reviews, which have been labelled as review and rating. So, it contains the reviews of customers along with the rating they have provided. The other thing about this data is that there is no categorization of the products. So, we can't get the idea that the reviewer is talking about which product. Moreover, it can't categorize the data like electronic items, Furniture, Cosmetics etc and check the sentiments in a particular section or product. But surely, with that much amount of large data, the system is going to get a lot of reviews and can feed it to the system and see interesting results [2].

3.2 Libraries

Some important libraries needed for the model to work are NumPy, Pandas, RE (Regular Expression), Count vectorizer, LR (Logistic Regression), accuracy score and Sklearn.

3.3 Data Pre-processing

For the text pre-processing, the model uses the RE (Regular Expression) library. Here, extract all the non-alpha-numeric and which are Capital characters. Then convert the entire review in lower case. Moreover, wherever there are non-alpha-numeric characters, special characters and non-ascii characters, it will be substituted with space as these don't play any part in sentiment analysis. So, now the system has ASCII characters for the review texts. Similarly, this will be implemented on training texts and testing texts too [9,11].

3.3.1 Sentiment Tokenization

Tokenization is one of the pre-processing techniques used in sentiment analysis for converting text into tokens before transforming it into vectors. It is very much easier to filter out the unnecessary tokens. We can do it by converting a document into paragraphs, or sentences into words. In this work, the reviews of customers are converted into words [17].

3.3.2 Stop words

They are commonly occurring words, which are not important for the context of the data. They also have no contribution for any deeper meaning or do not play any part in analysing the sentiment [10].

3.3.3 Normalization

Some words have the same meaning but different styles of writing. So, it needs to be processed correctly so that model doesn't treat it differently. For example, we can take "100" and "hundred" as equal or "Mango" and "mango" as equal.

3.4 Feature Extraction:

3.4.1 Count Vectorizer

It is one of the libraries from Sklearn feature extraction. The Count vectorizer uses a bag of words approach that ignores the text structures. It is efficient in only extracting the information through the word counts. So, first of all, it will transform each document in a vector form. And the input of the vector is taken as an occurrence count for each unique word from the document [16].

Now, suppose there are "m" documents in the corpus (for the collection of data, which we want to analyse) and "n" number of unique words in the document, the Count Vectorizer will transform itself to an "m*n" sparse matrix.

Count Vectorizer follows two steps to form a sparse matrix: fit and transform. In the fitting process, the vectorizer reads and counts the total number of words for the corpus. Then it assigns an index for each word [6]. The next step is the transformation of the fitted data. The occurrence of each unique word is counted in each document.

3.4.2 Tf-IDF Term Weighting

Some words like ‘a’, ‘is’, ‘the’ etc. are very less meaningful when it comes to analyzing the sentiment of any document. If we have to feed the direct count data to any classifier, the terms will shadow the frequencies of rarer or other interesting terms in the model. So, to re-weight the count features into floating-point values which are suitable by the classifier, the tf-idf transformer is commonly used [4].

Tf means ‘term-frequency’ while the Tf-IDF means ‘term frequency-inverse document frequency.

The formula used to compute the tf-idf of a term ‘t’ of a document ‘d’ in a set is:

$$\mathrm{tf}-\mathrm{idf}(\mathrm{t},\mathrm{d})=\mathrm{tf}(\mathrm{t},\mathrm{d})*\mathrm{idf}(\mathrm{t})$$

Equation 1 Tf-IDF Vectorizer Formula

and the idf is computed as:

$$\mathrm{idf}(\mathrm{t})=\log\left[\frac{\mathrm{n}}{\mathrm{df}(\mathrm{t})}\right]+1(\mathrm{if\,smooth}_{\mathrm{idf}}=\mathrm{False})$$

Equation 2 IDF Equation

Where n is the total number of documents in the document set and df(t) is the document frequency of t; the document frequency is the number of documents in the document set that contain the term ‘t’ [8].

3.5 Algorithm

The model uses the Logistic Regression Approach. This algorithm predicts the probability of an outcome that has two values. Linear Regression is not an appropriate algorithm for predicting the value of a binary variable, because it can predict the value outside a range (can be other than 0 or 1), but since logistic regression gives the binary result here, it is a very good approach for this system. Also, the logistic regression produces a curve that gives a limited value between 0 and 1. So, Logistic Regression is used when the dependable variable(target) is categorical, as in our case [11].

Logistic Regression can be of 3 types: Binary, Multinomial and Ordinal. The Binary gives the result as 2 possibilities, whereas Multinomial provides 3 or more categories without ordering like which of the food type is preferred more out of veg, non-veg or vegan. Lastly, Ordinal makes a category of 3 or more but considers the ordering too, like ranging from 1 to 5.

3.5.1 Decision Boundary

A threshold value is set to predict the class to where data belongs to. Say if in this Sentiment model, if predicted value >= 0.5, then it classifies as Negative sentiment otherwise Positive [7].

3.5.2 Accuracy Score

The accuracy score is another function of the SkLearn library, which computes the accuracy. In multilabel classification, this function returns the accuracy of the subset.

3.5.3 Training and Testing Split

The model uses the 'train_test_split' method, where the data is divided as 75% for the training set and 25% for the testing set.

3.5.4 C-Value

It is the method to regularize the model over the top of the data. It can be called the 'Inverse Strength of Regularization'. The system uses 5 different values of 'c' as 0.01, 0.05, 0.25, 0.5 and 1.0. As the value of 'c' increases, so does the regularization [5,18].

3.5.5 Regularisation

It is the method to improve performance over the unseen data. This method is used to avoid over-fitting. According to Ian Goodfellow, Regularization is a modification that we make to an algorithm(learning) that is intended to reduce its generalization error but not its training error [8].

In other words, we can say that this method is used to train models that generalize better on unseen data. This doesn't memorize the model, and does work well for any unseen data too. Thus, this improves the performance of unseen data.

3.6 Review Analysis

Now, comes the time to predict the data through the model which is created. One of the predictions displayed as "[1]" and when the test set was verified, it showed the same "[1]" for the same predicted piece of text. Now to show that particular review from the user, it read : "stunning even for the non-gamer this soundtrack was beautiful it paints the scenery in your mind so well i would recommend it even to people who hate video game music i have played the game chrono cross but out of all of the games i have ever played it has the best music it backs away from crude keyboarding and takes a fresher step with grate guitars and soulful orchestras it would impress anyone who cares to listen". As discussed earlier, "[1]" denotes the positive sentiment of a review while "[0]" denotes the negative sentiment of a review. Now you can read the above review and observe that it is giving positive feedback to the whatever product he bought.

3.7 Sentiment Classification

The pre-processed data is taken as input, put over the LR model built and generates an output. So, finally the model can classify the reviews as positive ([1]) and negative ([0])

sentiment. The output is generated in binary form. Thus, the motive of the model is fulfilled in the case of output generated, but we will further get into the result.

4. Results

The main aim of this paper is to generate fair results on the basis of the model built. No one wants to spend a lot of time analysing the quality and response from the customer toward their product. This system achieves an accuracy of about 80-90%, which is quite a good level of accuracy in such a large amount of data that it processes.

The experiments performed upon this model with the use of the CountVectorizer method gives a decent result. The testing is done with a range of different amounts of datasets taken at once. This result shows a little change in accuracy for each of the datasets taken into consideration. So now let's see the result for 10k, 100k, 300k and 500k of data when they are trained once with CountVectorizer and once with Tf-idfVectorizer.

Table 1: Result for 10k review data for different c-values

c-value	Tf-idf accuracy	CountVectorizer accuracy
0.01	71.12	83.88
0.05	82.64	85.28
0.25	84.44	85.88
0.5	84.96	85.76
1.0	86.12	85.48

Table 2: Result for 100k review data for different c-values

c-value	Tf-idf accuracy	CountVectorizer accuracy
0.01	71.52	87.54
0.05	83.54	88.65
0.25	85.78	88.21
0.5	86.85	88.56
1.0	87.23	88.24

Table 3: Result for 300k review data for different c-values

c-value	Tf-idf accuracy	CountVectorizer accuracy
0.01	85.8	88.81
0.05	87.7	89.57
0.25	89.2	89.63
0.5	89.51	89.29
1.0	89.74	89.08

Table 4: Result for 500k review data for different c-values

c-value	Tf-idf accuracy	CountVectorizer accuracy
0.01	86.52	89.22
0.05	88.31	89.97
0.25	89.59	89.94
0.5	90.03	89.81
1.0	90.21	89.69

Apart from the accuracy, this system shows the F1 Score and Precision Score [15]. The overall average precision score of both CountVectorizer and Tf-IDF Vectorizer is increasing smoothly and varying from 0.93 to 0.96, whereas the F1 score is varying from 0.86 to 0.90 and here the Tf-IDF Vectorizer is performing better than the CountVectorizer.

Table 5: F1 Score and Average Precision Score of CountVectorizer and Tf-IDF Vectorizer

		10k	100k	300k	500k
F1 Score	Tf-IDF	0.86	0.89	0.90	0.90
	CountVect	0.85	0.88	0.89	0.90
Average Precision	Tf-IDF	0.93	0.96	0.96	0.96
	CountVect	0.92	0.95	0.96	0.96

Now, look into the Precision vs Recall graph which the system generates from both the CountVectorizer and Tf-IDF Vectorizer:

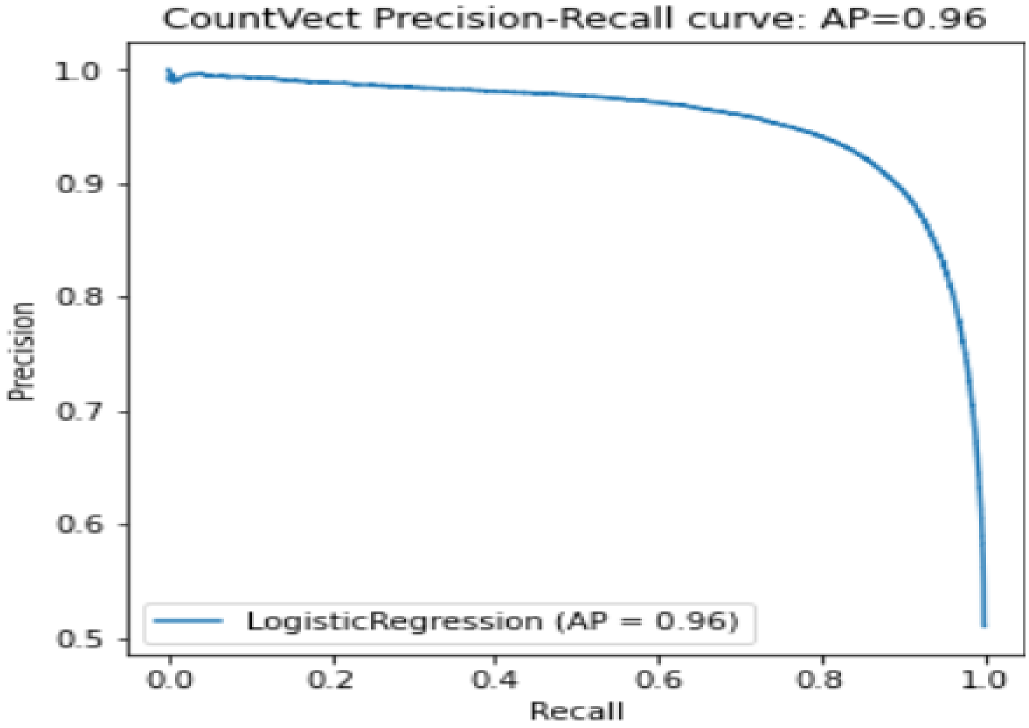


Figure 2**:** Precision vs Recall Curve for CountVectorizer

Figure 2 portrays the Count Vectorizer method from sklearn used for evaluating the performance of binary classification algorithms. It is often used in situations where classes are heavily imbalanced.

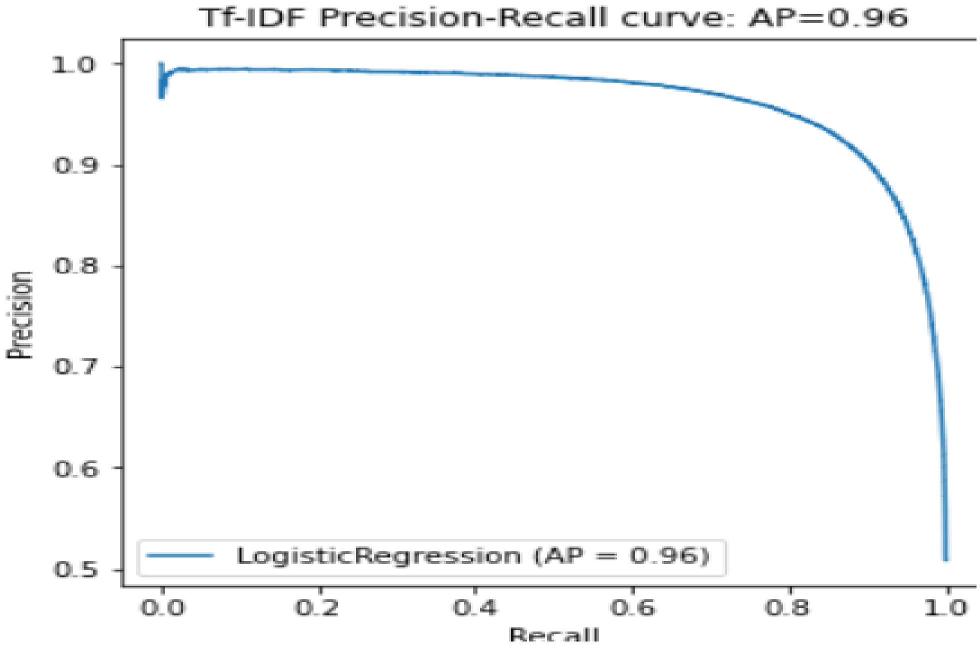


Figure 3**:** Precision-Recall Curve for Tf-IDF Vectorizer

Figure 3 denotes the performance of the classification.

These graphs help to know how good a model is predicting the positive class. Precision is referred to as positive predicted values. After this curve, have a look at the ROC Curve generated by the model. The ROC Curve means Receiver Operating Characteristics (ROC), which generally is a metric to evaluate the classifier output quality [13]. The ROC Curve features a True Positive rate on the Y-Axis while the False Positive rate on the X-axis.

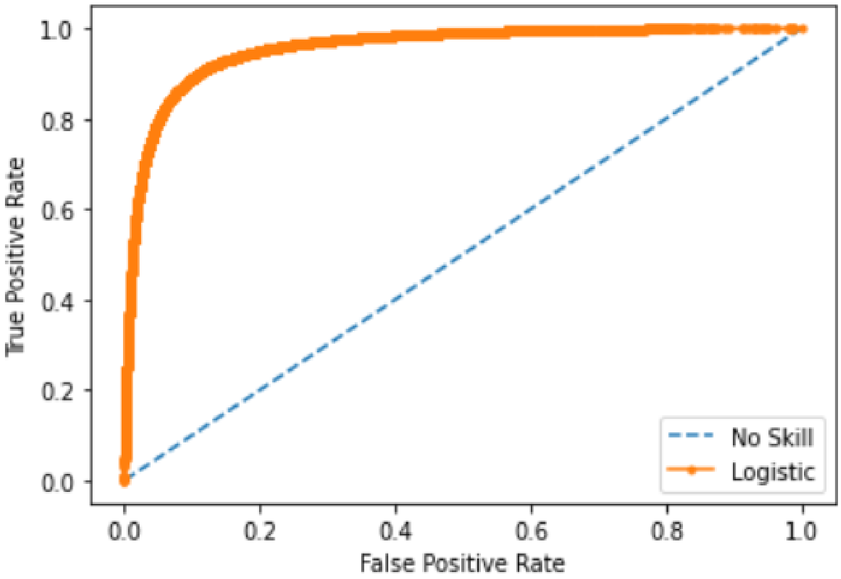


Figure 4: ROC Curve for Count Vectorizer

Figure 4 shows the true positive vs false Positive graph, which generally is a metric to evaluate the classifier output quality.

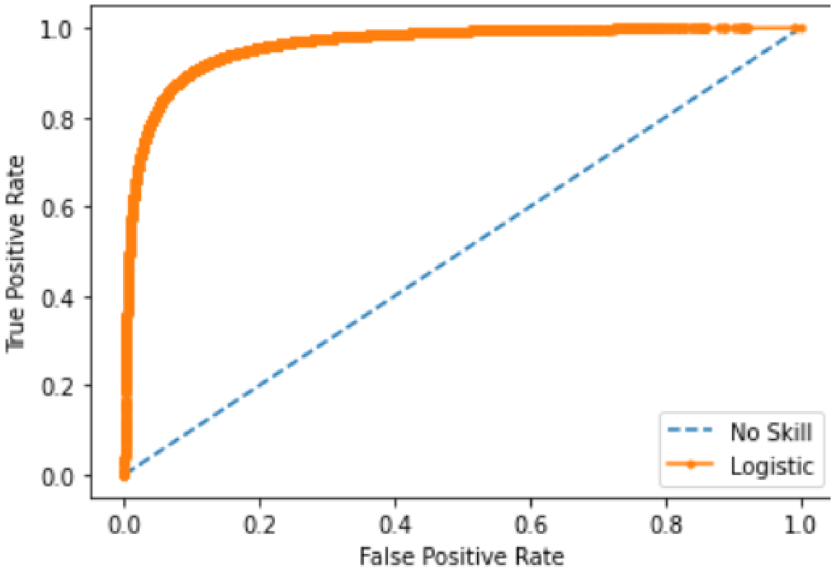


Figure 5: ROC Curve for Tf-IDF Vectorizer

Figure 5 shows the connection or trade-off between clinical sensitivity and specificity.

ROC curves are used in binary classification to study the output in the classifier. So, it is necessary to binarize the output if the output is not binary [14]. Given a threshold parameter T, the instance called as positive if X>T, where X is a continuous random variable, the TPR (True Positive Rate) is given by:

$$\mathrm{TPR(T)} = \int_{\mathrm{T}}^{\infty} \mathrm{f}_1(\mathrm{x})\mathrm{dx}$$

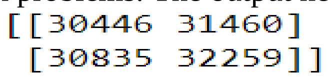
Equation 3 TPR (True Positive Rate) Equation

While the False Positive rate is denoted by:

$$\mathrm{FPR(T)} = \int_{\mathrm{T}}^{\infty} \mathrm{f}_0(\mathrm{x})\,\mathrm{dx}$$

Equation 4 FPR (False Positive Rate) Equation

The Confusion matrix for the CountVectorizer and the Tf-IDF vectorizer is also drawn in the module. The confusion matrix shows the performance measurement for ML classification problems. The output needs to be of two or more classes.



```
[[30446 31460]
 [30835 32259]]
```

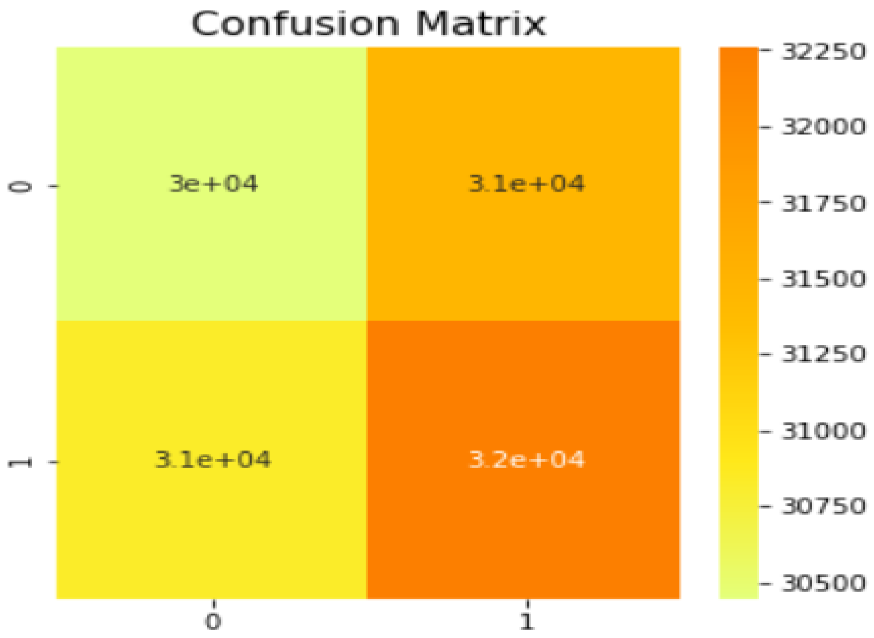
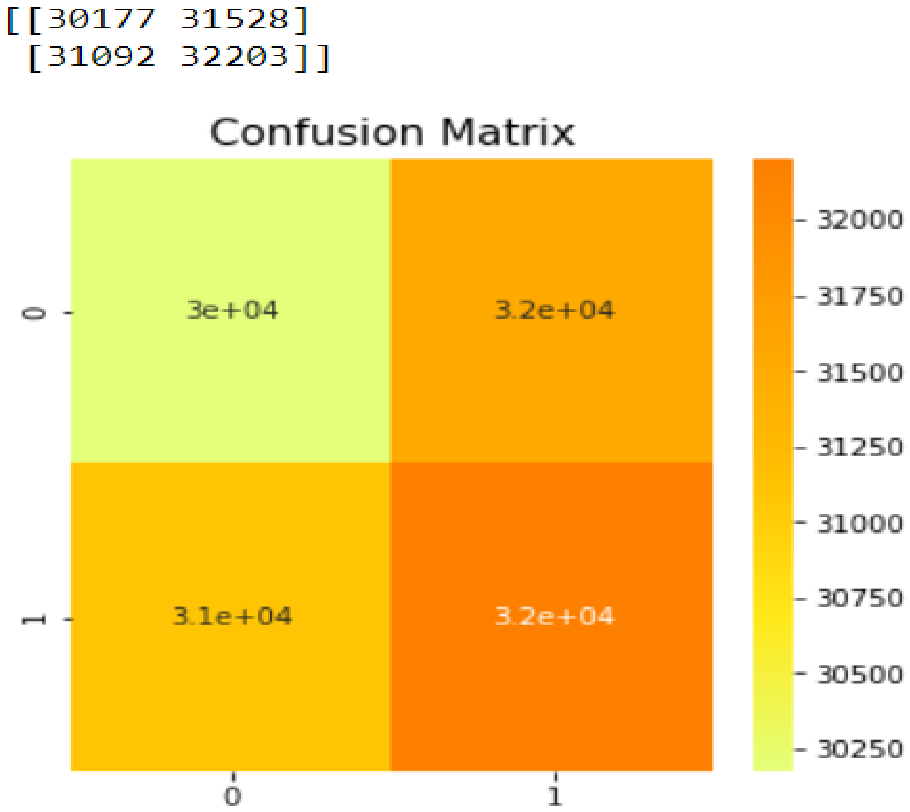


Figure 6: Confusion Matrix for CountVectorizer for 125k testing dataset

Here, one can observe that for total



```
[[30177 31528]
 [31092 32203]]
```

Figure 7: Confusion Matrix for Tf-IDF Vectorizer for 125k testing dataset

5. Conclusion

Sentiment analysis is one of the fastest-growing research areas, as it has a lot of scope and benefits. Though this research work started a decade back, still one can find a lot of improvements because the technology is improving day by day and so is the scope of sentiment analysis. This research work is showing a comparison of two methods of feature selection which are CountVectorizer and Tf-IDF vectorizer. One can easily conclude with the different results shown in the table that CountVectorizer is performing better than Tf-IDF Vectorizer, though, for a large amount of data, both the methods are achieving nearly 90% accuracy. With ranging from 83% to 90% of accuracy with different amounts of data, CountVectorizer is much more consistent as compared to Tf-IDF vectorizer. But when it comes to results, apart from accuracy, F1 Score and Precision-Recall score matters too. So, this work also presents the Precision vs Recall graph. Having a look at the F1 score, it can be observed that both the methods range from 0.85 and 0.90. Apart from that, the work defines the Average Precision Score, which ranges between 0.92 to 0.96 for both the methods. Lastly, the ROC Curve in Logistic Regression defines the best cut-off value for determining whether any new observation is a failure or a success [14]. The results also show the ROC Curve for both the methods. Similarly, a confusion matrix is also derived.

Lastly, this work showing the comparison of two unique methods through the logistic regression model can further be deployed through various other models and algorithms, which may provide better accuracy in predicting the reviews of people and accurately classifying those sentiments. Deep Convolutional Neural Network is one such method that can be one of the options for researchers. Many of the Deep Learning techniques have been a popular area. These algorithms automatically learn new complex features. Further developments in the field of sentiment analysis are surely possible.

Social media is one such platform where data is being generated uncontrollably. Data is generated continuously and so are the challenges. There is a possibility of new linguistic features being created since social media has no grammatical foundations. Hence, classification accuracy may decrease unless the model evolves and understand those texts. Sarcasm Analysis will be one of the most challenging topics in the upcoming time, as people also use sarcasm in reviews and texts nowadays. The verbal irony should be closely analysed.

As these days, public opinion matters a lot, so a vast area of research to predict those opinions or sentiments is there. This work will not only be confined to the reviews but also for each and every sector where any text sentiment is having some value for bringing some business.

References

1. Haque T, Saber N, Shah F (2018). Sentiment analysis on large scale Amazon product reviews. 2018 IEEE International Conference on Innovative Research and Development (ICIRD).
2. Shrestha N, Nasoz F (2019). Deep Learning Sentiment Analysis of Amazon.Com Reviews and Ratings. International Journal on Soft Computing, Artificial Intelligence and Applications. 2019;8(1):01-15.
3. Bhatt A, Patel A, Chheda H (2016). "Amazon Review Classification and Sentiment Analysis. ", International Journal of Computer Science and Information Technologies. 2015;6.
4. D'souza S, Sonawane K (2019). Sentiment Analysis Based on Multiple Reviews by using Machine learning approaches. Third International Conference on Computing Methodologies and Communication.
5. Kumari Singh A, Shashi M (2019). Vectorization of Text Documents for Identifying Unifiable News Articles. International Journal of Advanced Computer Science and Applications. 2019;10(7).
6. Sarlis S, Maglogiannis I (2020). On the Reusability of Sentiment Analysis Datasets in Applications with Dissimilar Contexts. IFIP Advances in Information and Communication Technology. 2020. 409-418.
7. Mäntylä M, Graziotin D, Kuutila M (2018). The evolution of sentiment analysis—A review of research topics, venues, and top cited papers. Computer Science Review. 2018; 27:16-32.
8. Nirag Bhatt, T. and Swarndeep Saket, A., (2021). Sentiment Analysis using Machine Learning Technique: A Literature Survey. International Research Journal of Engineering and Technology (IRJET), 07(12).
9. Kawade, D. and Oza, D. (2017). Sentiment Analysis: Machine Learning Approach. International Journal of Engineering and Technology, 9(3), pp.2183-2186.

10. Muhammad, A., Bukhori, S. and Pandunata, P., (2019). Sentiment Analysis of Positive and Negative of YouTube Comments Using Naïve Bayes – Support Vector Machine (NBSVM) Classifier. 2019 International Conference on Computer Science, Information Technology, and Electrical Engineering (ICOMITEE).
11. Rathi, M., Malik, A., Varshney, D., Sharma, R. and Mendiratta, S., (2018). Sentiment Analysis of Tweets Using Machine Learning Approach. 2018 Eleventh International Conference on Contemporary Computing (IC3),.
12. Raza, H., Faizan, M., Hamza, A., Mushtaq, A. and Akhtar, N., (2019). Scientific Text Sentiment Analysis using Machine Learning Techniques. International Journal of Advanced Computer Science and Applications,, 10(12), pp.157-165.
13. Sentiment analysis of reviews: Text Pre-processing [Internet]. Medium. 2021 [cited 20 August 2021]. Available from: https://medium.com/@annabiancajones/sentiment-analysis-of-reviews-text-pre-processing-6359343784fb.
14. Understanding Logistic Regression step by step [Internet]. Medium. 2021 [cited 20 August 2021]. Available from: https://towardsdatascience.com/understanding-logistic-regression-step-by-step-704a78be7e0a
15. Logistic Regression — Detailed Overview [Internet]. Medium. 2021 [cited 20 August 2021]. Available from: https://towardsdatascience.com/logistic-regression-detailed-overview-46c4da4303bc
16. Regularization for Logistic Regression: L1, L2, Gauss or Laplace? | KNIME [Internet]. KNIME. 2021 [cited 20 August 2021]. Available from: https://www.knime.com/blog/regularization-for-logistic-regression-l1-l2-gauss-or-laplace
17. Precision-Recall — scikit-learn 0.24.2 documentation [Internet]. Scikit-learn.org. 2021 [cited 20 August 2021]. Available from: https://scikit-learn.org/stable/auto_examples/model_selection/plot_precision_recall.html
18. Brownlee J (20).. ROC Curves and Precision-Recall Curves for Imbalanced Classification [Internet]. Machine Learning Mastery. 2021 [cited 20 August 2021]. Available from: https://machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-imbalanced-classification
19. F1 Score [Internet]. ritchieng.github.io. 2021 [cited 20 August 2021]. Available from: https://www.ritchieng.com/machinelearning-f1-score

***