Word clouds based on Different Methods of Word Vectorization on the Amazon shoes review dataset

I. Introduction

Using Word Cloud Vectorization, we analyze our data more intuitively. In this project, we will perform two-word vectorization methods and compare the word cloud generated by the data. Moreover, we will build 2 machine-learning models using the vectorization algorithm and compare their results. For this, we will use the US Shoes Review Dataset of Amazon provided by TensorFlow.

II. About the Dataset

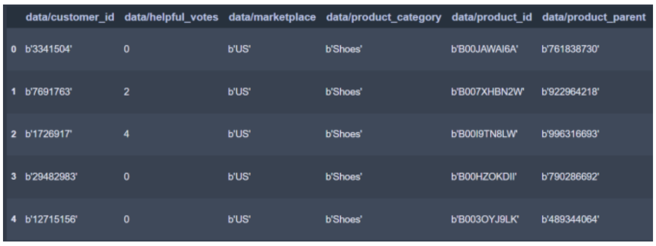
The dataset for this project is collected from TensorFlow datasets [1]. This US shoes dataset is the subset of the whole TensorFlow dataset. Since this dataset has more than 130 million rows, we will take only the last 20,000 rows for this project as it will be heavier to train the model. The dataset description is as follows:

a) customer_id: unique identifier for each customer
b) helpful_votes: number of helpful votes for the review
c) marketplace: the Amazon marketplace where the review was written (US)
d) product_category: category of the product (Shoes)
e) product_id: product unique identifier
f) product_parent: parent product identifier
g) product_title: title of the product
h) review_body: body part of the review
i) review_date: date of the review
j) review_headline: headline of the review
k) review_id: review unique identifier
l) star_rating: the rating by customer (1-5 stars)
m) total_votes: total number of votes for the review
n) verified_purchase: whether the product is purchased from Amazon and the review is verified.
o) vine: whether customers received free products in exchange for reviews.

III. Data Preparation

A. Querying the Data

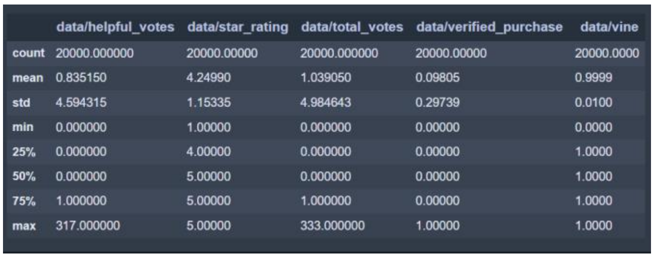
It consists of 20,000 rows and 15 feature columns.



	data/customer_id	data/helpful_votes	data/marketplace	data/product_category	data/product_id	data/product_parent
0	b'3341504'	0	b'US'	b'Shoes'	b'B00JAWAI6A'	b'761838730'
1	b'7691763'	2	b'US'	b'Shoes'	b'B007XHBN2W'	b'922964218'
2	b'1726917'	4	b'US'	b'Shoes'	b'B00I9TN8LW'	b'996316693'
3	b'29482983'	0	b'US'	b'Shoes'	b'B00HZOKDII'	b'790286692'
4	b'12715156'	0	b'US'	b'Shoes'	b'B003OYJ9LK'	b'489344064'

Fig. 1. Sample of Amazon US Shoes Review Dataset

B. Descriptive Analysis



	data/helpful_votes	data/star_rating	data/total_votes	data/verified_purchase	data/vine
count	20000.000000	20000.00000	20000.000000	20000.00000	20000.0000
mean	0.835150	4.24990	1.039050	0.09805	0.9999
std	4.594315	1.15335	4.984643	0.29739	0.0100
min	0.000000	1.00000	0.000000	0.00000	0.0000
25%	0.000000	4.00000	0.000000	0.00000	1.0000
50%	0.000000	5.00000	0.000000	0.00000	1.0000
75%	1.000000	5.00000	1.000000	0.00000	1.0000
max	317.000000	5.00000	333.000000	1.00000	1.0000

Fig. 2. Descriptive analysis of the dataset

According to the dataset description, the evaluations had an average of 0.84 helpful votes and a star rating of 4.25 out of 5. The median number for helpful and total votes was zero, showing that most evaluations received no votes. Furthermore, most evaluations were verified purchases, not vine programme participants.

IV. Data Cleaning

We can see from (Fig.1) that the column header and the dataset have errors. So, we are modifying the original column names to make them more readable and relevant to our analysis. Moreover, in the dataset, we can see that most of the rows start with the value 'b' prefix. The error related to 'b' in the data frame columns is caused by the fact that some of the values in the columns are byte strings. These byte strings need to be converted to regular strings. We applied the regular expression to columns 'helpful_votes', 'star_rating', 'total_votes', 'verified_purchase', and 'vine'.

V. Missing Values Handling

For checking the missing values, we applied the compute ration formula to determine whether Rm = 0 for each feature. This dataset does not contain null values.

VI. Feature Selection

While checking each column, some columns are optional. This dataset is from the US, so we don’t need a marketplace column. Similarly, the product_category column is also unnecessary as it has all values as shoes. The column review_headline consists of the heading about what is inside the body column. So we can only consider review_column and remove all other unnecessary columns irrelevant to our analysis. After selecting features, six columns are removed from the dataframe.

VII. Text Preprocessing

For text preprocessing, the review_body column is used and passed through all the natural language processing steps.

A. *Convert String to Lowercase:*

All the strings in each row are converted to lowercase in this step.

B. *Tokenization:*

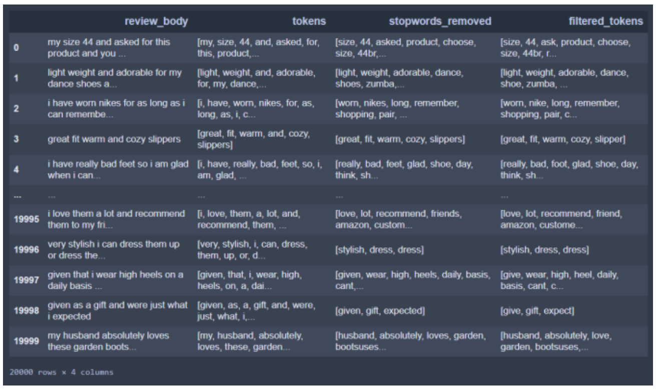
Each lowercase word is tokenized using the nltk package and stored in the tokens column in the dataframe.

C. *Stopwords Removal:*

Stop words like ‘the’, ‘this’, ‘those’, etc., are removed from each tokenized row for better model performance.

D. *Lemmatization and POS tagging:*

Here, after stop words removal, we applied POS tagging before applying lemmatization to each token and defined a function to map treebank POS tags to their corresponding WordNet POS tags. The resulting lemmatized words are stored in a data frame column.



	review_body	tokens	stopwords_removed	filtered_tokens
0	my size 44 and asked for this product and you ...	[my, size, 44, and, asked, for, this, product,...	[size, 44, asked, product, choose, size, 44br,...	[size, 44, ask, product, choose, size, 44br, r...
1	light weight and adorable for my dance shoes a...	[light, weight, and, adorable, for, my, dance,...	[light, weight, adorable, dance, shoes, zumba,...	[light, weight, adorable, dance, shoe, zumba, ...
2	i have worn nikes for as long as i can remembe...	[i, have, worn, nikes, for, as, long, as, i, c...	[worn, nikes, long, remember, shopping, pair, ...	[worn, nike, long, remember, shopping, pair, c...
3	great fit warm and cozy slippers	[great, fit, warm, and, cozy, slippers]	[great, fit, warm, cozy, slippers]	[great, fit, warm, cozy, slipper]
4	i have really bad feet so i am glad when i can...	[i, have, really, bad, feet, so, i, am, glad, ...	[really, bad, feet, glad, shoe, day, think, sh...	[really, bad, foot, glad, shoe, day, think, sh...
...	...	...	...	...
19995	i love them a lot and recommend them to my fri...	[i, love, them, a, lot, and, recommend, them, ...	[love, lot, recommend, friends, amazon, custom...	[love, lot, recommend, friend, amazon, custome...
19996	very stylish i can dress them up or dress the...	[very, stylish, i, can, dress, them, up, or, d...	[stylish, dress, dress]	[stylish, dress, dress]
19997	given that i wear high heels on a daily basis ...	[given, that, i, wear, high, heels, on, a, dai...	[given, wear, high, heels, daily, basis, cant,...	[give, wear, high, heel, daily, basis, cant, c...
19998	given as a gift and were just what i expected	[given, as, a, gift, and, were, just, what, i,...	[given, gift, expected]	[give, gift, expect]
19999	my husband absolutely loves these garden boots...	[my, husband, absolutely, loves, these, garden...	[husband, absolutely, loves, garden, bootsuses...	[husband, absolutely, love, garden, bootsuses,...

20000 rows × 4 columns

Fig. 3. Dataframe after text pre-processing

After applying all the text preprocessing steps, we have obtained clean tokens, which are appended in the data frame column, as shown in Fig. 3.

VIII. Vectorization

After the data text preprocessing steps, the next step is to apply the word vectorization methods to convert the text data into a numerical format that can be used as input to the machine learning model. The two-word vectorization methods that will be compared in this analysis are CountVectorizer and TF-IDF Vectorizer. We generated word clouds for visualizing text data using these vectorizers. As we are dealing with a large corpus of data with many common words, so TF-IDF might perform better.

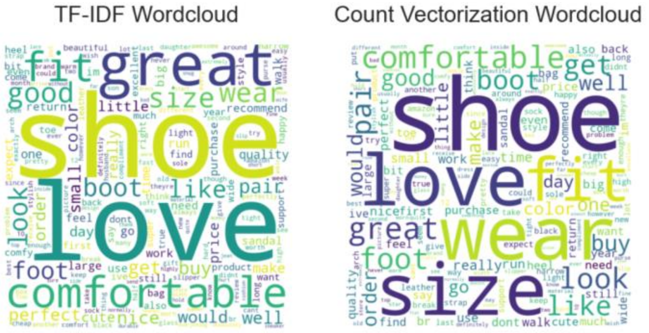


Fig. 4. TF-IDF and Count Vectorizer word cloud comparison

From fig 4, we can see the word clouds for both methods. Here, We can see word clouds visualization for text data, but we cannot use it as the basis for evaluating different vectorizers' performance. Furthermore, we will use machine learning algorithms for a more reliable result.

IX. Data Modelling

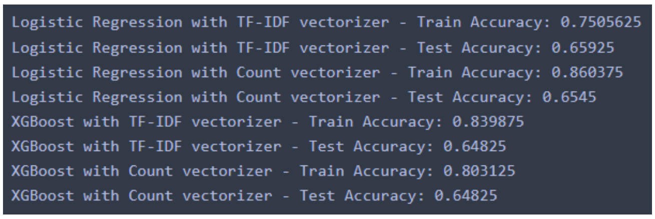
A. *Building Pipeline:*

We will build a pipeline by applying TF-IDF and Count Vectorization to machine-learning algorithms. The Data is split into 20% test data and 80% train data.

B. *Supervised Algorithm:*

Here, We trained two logistic regression models and two XGBoost models. We used both vectorizers for each sort of model, yielding four models in total. We created pipelines using the Scikit-learn library to combine the vectorizer and model steps. We fit each model to the training data and evaluated their performance on the test data using accuracy scores.

X. Results



```
Logistic Regression with TF-IDF vectorizer - Train Accuracy: 0.7505625
Logistic Regression with TF-IDF vectorizer - Test Accuracy: 0.65925
Logistic Regression with Count vectorizer - Train Accuracy: 0.860375
Logistic Regression with Count vectorizer - Test Accuracy: 0.6545
XGBoost with TF-IDF vectorizer - Train Accuracy: 0.839875
XGBoost with TF-IDF vectorizer - Test Accuracy: 0.64825
XGBoost with Count vectorizer - Train Accuracy: 0.803125
XGBoost with Count vectorizer - Test Accuracy: 0.64825
```

Fig. 5. Train and test results for each combination of model

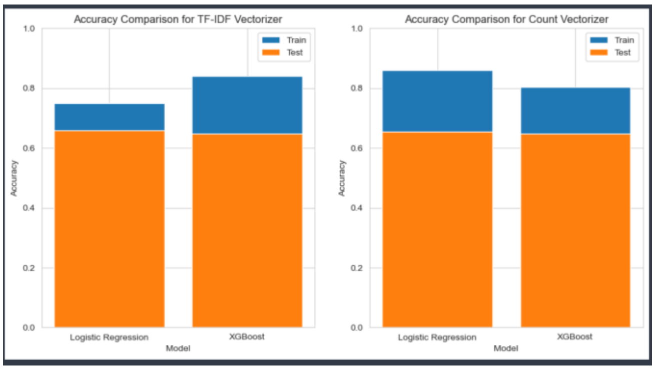


Fig. 6. TF-IDF and Count Vectorizer accuracy comparison

From the above fig 5 and fig 6, we discovered that models using the TF-IDF vectorizer outperformed the Count vectorizer regarding test accuracy. This is because the TF-IDF vectorizer assigns higher weights to more informative and less frequent words in the dataset while assigning lower weights to more familiar terms, allowing the model to concentrate on the more essential words for prediction. All the models, however, demonstrate a disparity between training and test accuracy, indicating that the models are overfitting the training data. This suggests that the models learned the training data too well, causing them to perform poorly on new data.

XI. Data Visualization

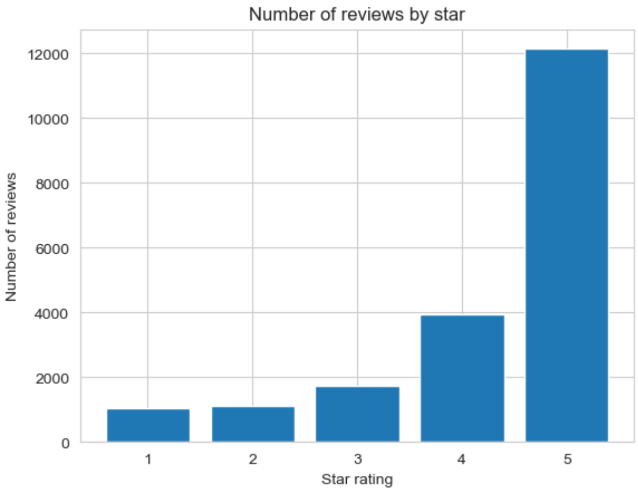


Fig. 7. Number of reviews based on the star rating

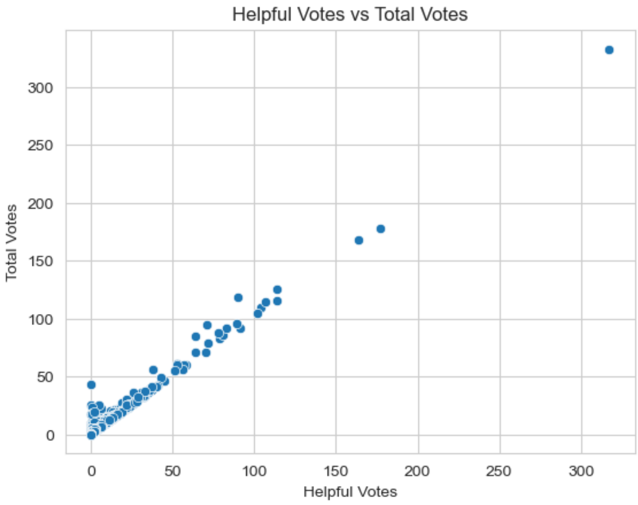


Fig. 8. Scatterplot of Helpful Votes and Total Votes column

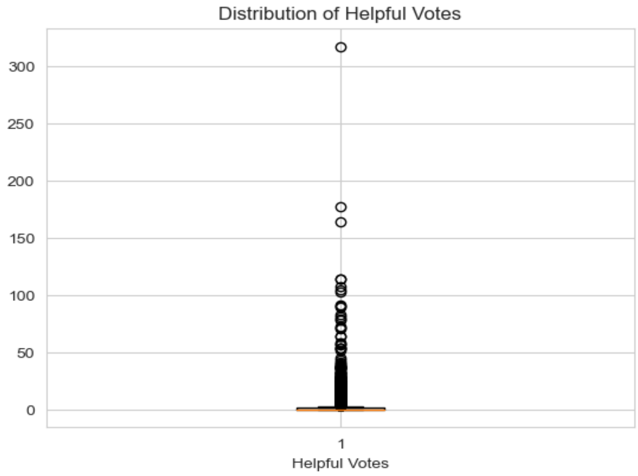


Fig. 9. Box plot of Helpful Votes column

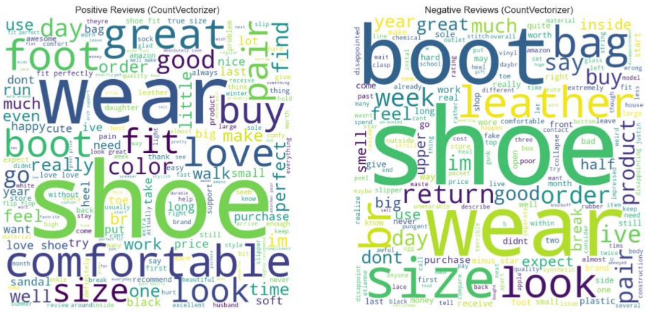


Fig. 10. Positive and Negative Reviews word cloud for Count Vectorizer

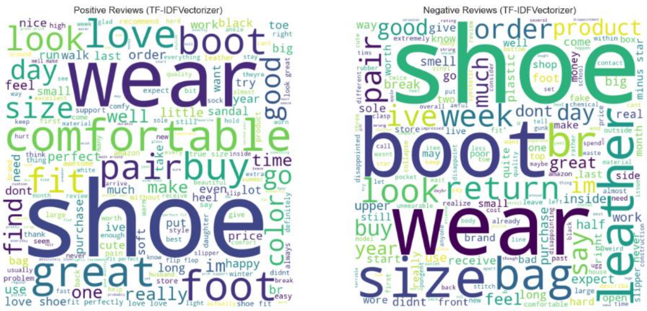


Fig. 11. Positive and Negative Reviews word cloud for TF-IDF Vectorizer

Conclusion

In this project, we demonstrated the usefulness of natural language processing techniques and word vectorization methods in analyzing large text datasets. Two vectorization methods, Count Vectorizer and TF-IDF Vectorizer, were used to preprocess and analyze the US Shoes Review Dataset of Amazon. We also built two machine-learning models using vectorization algorithms and compared their results. Moreover, we visualized the most frequent words in the review using the word clouds generated by these vectorization methods.

References

[1] TensorFlow Datasets, "Amazon US Reviews/Shoes_v1_00," [Online].Available: https://www.tensorflow.org/datasets/catalog/amazon_us_reviews#amazon_us_reviewsshoes_v1_00