CSE508 Information Retrieval

Winter 2024
Assignment-3

Name: Aniketh   29th March   Roll No: 2020360

Dataset Overview

The dataset contains a total of X rows, each representing a product review. The dataset includes several features such as 'reviewText', 'overall' rating, and 'asin' (Amazon Standard Identification Number), among others. I filter by the product ‘Turntable’.

Preprocessing Steps

Handling Missing Values:

- Missing values in the 'reviewText' column were replaced with empty strings to ensure compatibility with text-based models.
- Any other missing values in other columns were handled appropriately based on the context.

Handling Duplicates:

- Duplicate rows, if any, were identified and removed to ensure each review is unique.

Other Preprocessing Steps:

- Text normalization techniques were applied to the 'reviewText' column, including converting text to lowercase and removing special characters.
- The 'overall' rating column was used to create a new target variable 'Rating_Class' by categorizing ratings into 'Good', 'Average', and 'Bad' based on predefined criteria.
- a. Removing the HTML Tags.
- b. Removing accented characters.
- c. Expanding Acronyms. I did not perform any acronym expansion as this would have created some confusion for my products and brands names
- d. Removing Special Characters
- e. Lemmatization
- f. Text Normalizer

EDA

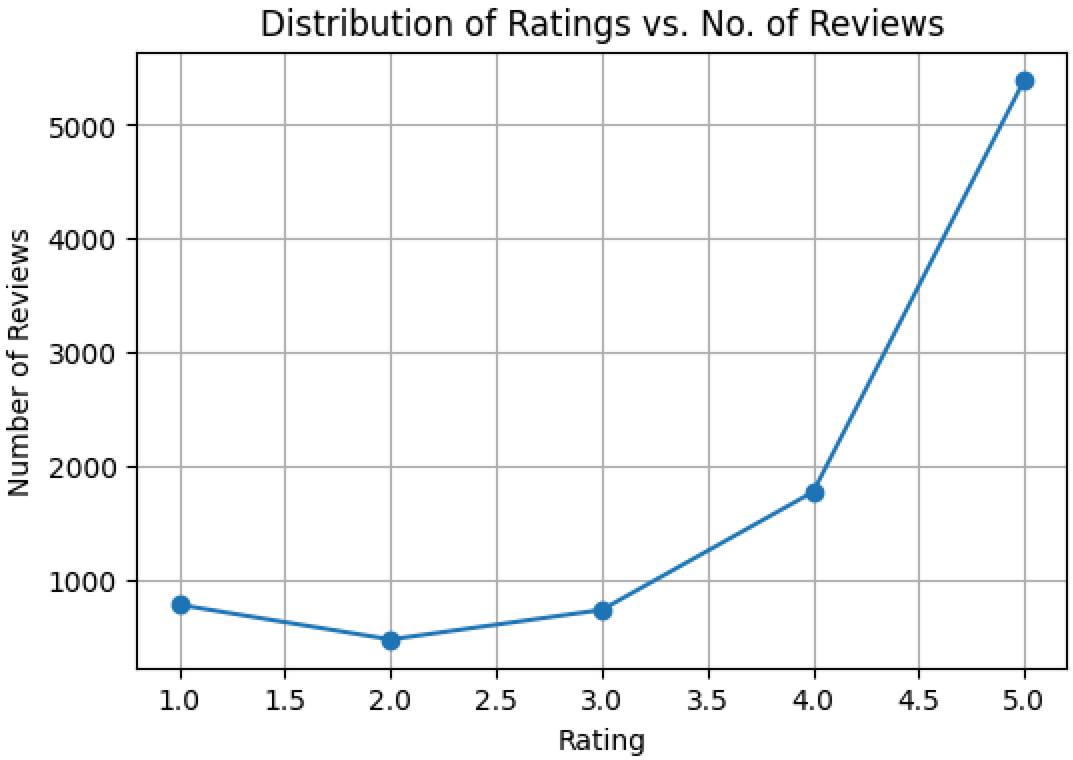
Top 20 most reviewed brands:

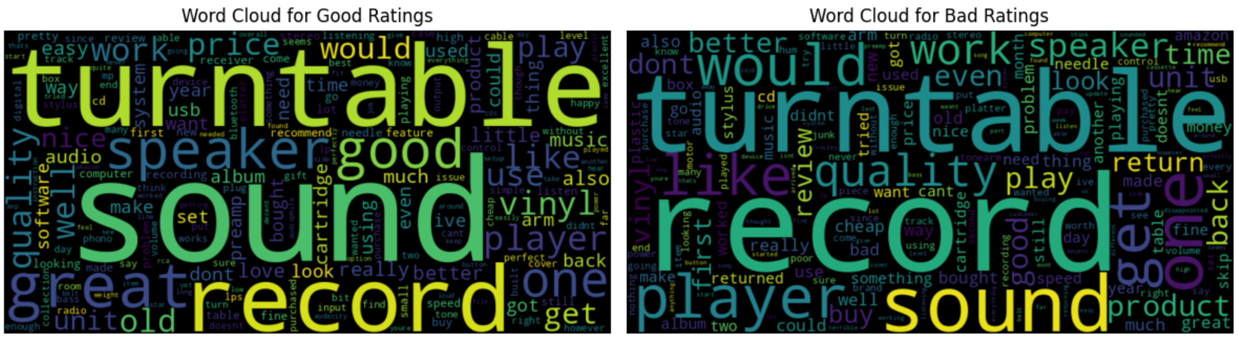
Jensen	1303
Audio-Technica	1149
WOCKODER	797
Pyle	795
Crosley	758
Victrola	462
ION Audio	450
Sony	320
Micca	305
Electrohome	254
Teac	228
BoxLegend	166
Pro-Ject	147
GOODNEW	135
Numark	115
Sylvania	115
Ion	114
Pioneer	111
1byone	102

Turntable Toys	96

Top 20 least reviewed brands:

Accessory Genie	10
TDK	10
LuguLake	8
Vibe	8
MUSITREND	8
Milestone Av Technologies	7
PAXCESS	7
Intellitouch	7
Thorens	7
CD Supply	6
Craig Electronics	6
Miles Kimball	5
TacPower	5
jWIN	5
GE	5
Grace Digital	5
it.innovative technology	5
Empire Scientific	5
UPBRIGHT	5
Sharp	5





Year with maximum reviews: 2015

Year with highest number of customers: 2015

Number of reviews for the year with the highest number of customers: 1198

Number of unique users for the year with the highest number of customers: 1185

Feature Engineering

I used a word2vec model to create tokens and then embeddings for the words in the corpus, using libraries from nltk.

Using this feature engineering I performed analysis to identify words similar to good/bad to test out the word2vec model.

Most similar words to 'good': [('amazing', 0.9634362459182739), ('ok', 0.9464921355247498), ('fantastic', 0.9458152651786804), ('expected', 0.9328287243843079), ('impressed', 0.9251856207847595)]

Most similar words to 'bad': [('point', 0.954521656036377), ('expecting', 0.9475014209747314), ('fair', 0.945181131362915), ('hype', 0.9437717199325562), ('considering', 0.9437171220779419)]

Data Split

Post this stage I created a train/test split for the data using scikit-learn library, and also performed some steps to adapt the data for training. I created a category column and applied a custom function on it to transform the ratings values to 3 categories. And filled nan values withe mpty reviews.

Models

Post this I created 5 ML Models to train an test the accuracy on our dataset. For all these models I resorted to using tfidf vectorizer as this gave me more efficient and better results.

1. Logistic regression

```
              precision    recall  f1-score   support

     Average       0.18      0.24      0.21       170
         Bad       0.61      0.55      0.58       340
        Good       0.88      0.88      0.88      1783
```

```
    accuracy                           0.78      2293
   macro avg       0.56      0.55      0.55      2293
weighted avg       0.79      0.78      0.79      2293
```

2. Multinomial Naive Bayes

```
              precision    recall  f1-score   support

     Average       0.00      0.00      0.00       170
         Bad       1.00      0.02      0.03       340
        Good       0.78      1.00      0.88      1783

    accuracy                           0.78      2293
   macro avg       0.59      0.34      0.30      2293
weighted avg       0.75      0.78      0.69      2293
```

3. XG Boost Classifier

```
              precision    recall  f1-score   support

     Average       0.40      0.10      0.16       170
         Bad       0.80      0.47      0.59       340
        Good       0.85      0.98      0.91      1783

    accuracy                           0.84      2293
   macro avg       0.69      0.52      0.56      2293
weighted avg       0.81      0.84      0.81      2293
```

4. Support Vector Classifier

```
              precision    recall  f1-score   support

     Average       1.00      0.02      0.03       170
         Bad       0.97      0.09      0.16       340
        Good       0.79      1.00      0.88      1783

    accuracy                           0.79      2293
   macro avg       0.92      0.37      0.36      2293
weighted avg       0.83      0.79      0.71      2293
```

5. Custom Transformer Model

```
              precision    recall  f1-score   support

           0       0.48      0.12      0.20       170
           1       0.78      0.68      0.73       340
           2       0.89      0.97      0.93      1783

    accuracy                           0.87      2293
   macro avg       0.72      0.59      0.62      2293
weighted avg       0.84      0.87      0.84      2293
```