Uluslararası İleri Doğa Bilimleri ve Mühendislik Araştırmaları Dergisi
Sayı 7, S. 263-268, 7, 2023

Araştırma Makalesi

IJANSER

https://as-proceeding.com/index.php/ijanser
ISSN: 2980-0811

International Journal of Advanced Natural Sciences and Engineering Researches
Volume 7, pp. 263-268, 7, 2023

Research Article

Enhancement of Sentiment Analysis Classification Performance of Spotify Reviews through NLP-based Approaches

Buğra Kağan KAYHAN[1], Murat ŞİMŞEK[2]

[1] *Faculty of Engineering, Software Engineering, Ostim Technical University, Ankara, Turkey*
[2]*Faculty of Engineering, Artificial Intelligence Engineering, Ostim Technical University, Ankara, Turkey*

(murat.simsek@ostimteknik .edu.tr) Email of the corresponding author

(Received: 16 August 2023, Accepted: 29 August 2023)

(1st International Conference on Recent and Innovative Results in Engineering and Technology ICRIRET 2023, August 16-18, 2023)

ATIF/REFERENCE: Kayhan, B. K. & Şimşek, M. (2023). Enhancement of Sentiment Analysis Classification Performance of Spotify Reviews through NLP-based Approaches, *International Journal of Advanced Natural Sciences and Engineering Researches*, 7(7), 263-268.

Abstract **–** In this study, sentiment analysis of Spotify reviews was conducted utilizing Natural Language Processing (NLP) techniques. Specific preprocessing steps were implemented to enhance the performance of sentiment classification, aiming to achieve superior outcomes compared to NLP-based solutions. These preprocessing procedures hold significant importance in effectively categorizing emotions. The resultant sentiment analysis categorized sentiments into three classes: negative, neutral, and positive. These derived classes were subjected to a comparative analysis employing machine learning methodologies. The comparative assessment revealed that the Xgboost method exhibited a more successful classification performance compared to other approaches. To address data imbalance within the dataset, various handling imbalance methods were employed and juxtaposed against one another. Through this investigation, the study contributes to a more comprehensive understanding of sentiment analysis of Spotify reviews using NLP, shedding light on optimal preprocessing strategies and effective machine learning techniques for sentiment classification in this context.

Keywords – NLP, Machine Learning, Classification, Sentiment Analysis, Handling Imbalance Dataset

I. INTRODUCTION

In the digital age, the proliferation of online platforms has led to an unprecedented volume of user-generated content, reflecting a wide spectrum of opinions, emotions, and sentiments. One such platform that has gained immense prominence is Spotify, a leading music streaming service that boasts millions of active users worldwide. As users engage with the platform, they often express their experiences, preferences, and critiques through written reviews, offering a valuable repository of sentiment-rich data. The analysis of sentiment within Spotify reviews not only provides insights into user satisfaction and dissatisfaction but also offers a unique lens through which to understand evolving musical preferences and trends. This academic paper delves into the realm of sentiment analysis applied to Spotify reviews, aiming to explore the nuanced interplay between user sentiments and the dynamic landscape of music consumption in the digital era. Through rigorous analysis and interpretation of sentiment-laden textual data, this study contributes to a deeper comprehension of the intricate relationship between technology, music, and human emotion.

Boasting an extensive catalog and accessible across diverse platforms including desktops, mobile devices, game consoles, and automobiles, Spotify stands as a prominent music streaming service. Holding a dominant position in the industry, they command a user base exceeding 75 million, out of which over 30 million are subscribed, underscoring their leadership in the market[1].

The integration of streaming platforms as a principal conduit within the music industry's framework has become an undeniable inevitability. As articulated by both Jones (2017) and Paradise (2014), these platforms not only serve to mend the rift instigated by online piracy but have also triumphantly reshaped the musical domain. The global populace's reliance on the internet has further fortified the ascendancy of streaming platforms as the preferred medium of distribution. Garnering a substantial subscription base of 434 million users (IFPI, 2021), not to mention an even more extensive contingent of non-paying enthusiasts, music streaming stands as one of the most auspicious realms for business development within the sphere of entertainment [2-3].

Embedded within a given context, sentiment analytics endeavors to discern and unveil the underlying perspectives or emotional inclinations encapsulated within a piece of discourse, expression, or any form of communication. The diverse linguistic tapestry woven by humanity across the globe serves as a pivotal conduit for the articulation of thoughts and sentiments. Irrespective of the language employed, sentiments are inherently intertwined with verbal communication, encompassing a spectrum ranging from affirmative to adverse or even impartial orientations[4].

In the realm of contemporary business landscapes, the proliferation of reviews, numbering in the thousands and millions, has rendered the manual scrutiny of each individual customer critique an insurmountable task over time. In addressing this formidable challenge, sentiment analytics emerges as a pivotal solution, offering the capacity to comprehensively assess voluminous troves of reviews and thereby inform decision-making processes pertaining to future advancements. By harnessing real-world evidence derived from extensive data sets, sentiment analytics transcends the limitations inherent to reliance on diminutive data samples. This endeavor is significantly fortified through the symbiotic integration of natural language processing and machine learning (ML) methodologies, ushering in a potent amalgamation that empowers the systematic analysis of these reviews [5].

This study aims to conduct an artificial intelligence-based sentiment analysis on Spotify reviews using an acquired dataset. Within the framework of this NLP-driven investigation, a comprehensive end-to-end solution is presented, focusing on the enhancement of classification performance metrics for machine learning algorithms. The study offers an integrated approach that spans from preprocessing to classification, striving to amplify the efficacy of sentiment analysis within the context of music reviews on the Spotify platform.

II. Materials and Method

A. The dataset

Spotify is one of the largest music streaming service providers, with over 422 million monthly active users, including 182 million paying subscribers, as of March 2022. Some of them don't hesitate to share their experience using this application along with the given rating to denote how satisfied they are with the Application. The dataset is taken from the Spotify App Reviews study on the Kaggle site [6]. This dataset was collected from Google Store by M Faarisul ILMI by scraping method.

The dataset consists of 61594 columns and 5 columns. The column is named Time-submitted, Review, Rating, Total_thumbsup, Reply in order. The columns are explained below:

- *Time_submitted:* At what time the review was submitted
- *Review:* Review text
- *Rating:* Given score (1-5)
- *Total_thumbsup:* How many people found the review helpful
- *Reply:* Review reply

Time_submitted and Reply columns, except for the columns that are the basis of the problem, are discarded.

B. *Exploratory Data Analysis*

Basic exploratory data analysis was performed on the data. In this context, missing data analysis was performed and no missing data were observed.

In the visualization made about our target column, "Rating", there are comments with up to 1-5 stars.

In Fig.1, the number of comments with 1-5 stars in the Rating column is visualized.

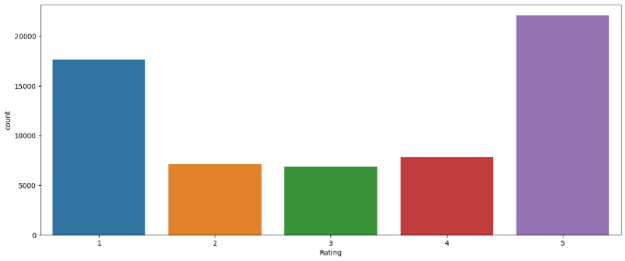

Fig. 1 Number of comments with 1-5 stars in the Rating column

This target, which will be the subject of sentiment analysis, is classified as negative-neutral-positive according to the number of stars in the column. 1-2 stars are labeled "Negative", 3 stars are "Neutral", 4-5 stars are labeled "Positive". The graphical representation of this transformation is given in Fig.2.

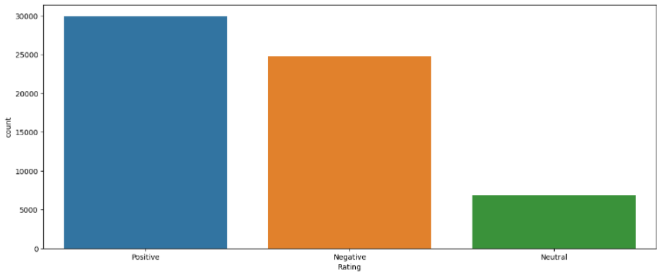

Fig. 2 Negative, Neutral, Positive class transformation

C. *Text Prepocessing*

Certain preprocessing has been used to perform sentiment analysis from the comments. The NLTK library was used for this. NLTK is a leading platform for building Python programs to work with human language data (Natural Language Processing). NLTK is intended to support research and teaching in NLP or closely related areas, including empirical linguistics, cognitive science, artificial intelligence, information retrieval, and machine learning.

The following steps were followed for preprocessing.

1. Lowercase
2. Remove Punctuations
3. Tokenization
4. Stopword Removal
5. Lemmatization

Converting text to lowercase is a common preprocessing step in NLP, and it is often done in conjunction with expanding contractions. Converting all text to lowercase helps to normalize the text, which means that different variations of the same word will be treated as the same token. For example, if "dog" and "Dog" were not normalized to the same case, they would be treated as different tokens even though they have the same meaning. When we convert text to lowercase, we reduce the size of our vocabulary by collapsing all variants of a word into a single token. This is particularly important for applications with limited resources, such as mobile devices or embedded systems. Many NLP tasks are case sensitive, meaning that "dog" and "Dog" would be treated as different words. However, in many cases, we do not want the model to be sensitive to case because it can introduce unnecessary noise in the data. For example, if we are trying to classify documents by topic, we likely do not care if "Dog" is capitalized or not. Converting all text to lowercase helps to ensure consistency in the data, which can improve the accuracy and reliability of NLP models. For example, if some text is in uppercase and some is in lowercase, it could be more difficult for the model to identify patterns and make accurate predictions.

Second step is Removing Punctuation. Removing punctuations is a common preprocessing step that is often done after expanding contractions and converting text to lowercase. There are several reasons why this is done:

Noise Reduction: Punctuations such as periods, commas, and semicolons add noise to the data, making it more difficult for NLP models to identify patterns and extract meaningful information. By removing punctuations, we can reduce the amount of noise in the data and improve the accuracy of NLP models.

Tokenization: Punctuation marks are often used to separate words and phrases, which means that if we keep them in the text, they will be treated as separate tokens. By removing punctuations, we can improve the accuracy of tokenization and ensure that words are correctly identified as separate tokens.

Normalization: Punctuations are not typically informative and can create unnecessary variability in the data. Removing punctuations can help to normalize the text and reduce the amount of variability in the data.

Consistency: Similar to converting text to lowercase, removing punctuations can help to ensure consistency in the data. By removing punctuations, we can reduce the number of variations of the same word or phrase and improve the accuracy and reliability of NLP models.

The next step is Removing stop words. Stop words are words that are considered to be common or uninformative and are typically removed from the text before analysis. Examples of stop words include "the", "and", "a", "an", "in", "of", and "to".

The last operation is Lemmatization. Lemmatization is the process of reducing words to their base or root form, known as a lemma. The goal of lemmatization is to transform words into their canonical or dictionary form, which helps to reduce the dimensionality of the data and improve the accuracy of NLP models. For example, the lemma of the word "running" is "run", and the lemma of the word "mice" is "mouse". By reducing all variations of a word to its base form, we can treat different forms of the same word as a single term, which can improve the accuracy of NLP models.

Lemmatization is different from stemming, another technique for reducing words to their root form. While stemming involves simply removing the suffixes from words to reduce them to a common root form, lemmatization involves using a dictionary or morphological analysis to identify the base form of each word [7].

In NLP, lemmatization is often used as a preprocessing step before other analyses, such as text classification or sentiment analysis. It helps to reduce the noise in the data by standardizing words to their base forms, which in turn helps to improve the accuracy of NLP models. Additionally, lemmatization can help to address the problem of sparsity, which can occur when different forms of the same word are treated as separate terms, leading to overfitting and reduced accuracy, lemmatization is a useful technique for reducing the dimensionality of the data and overall improving the accuracy and efficiency of NLP models.

Word Cloud was used to visualize the frequency and importance of the words used in the comments in the "Review" column in the dataset. Word Cloud is a data visualization technique used for representing text data in which the size of each word indicates its frequency or importance. Significant textual data points can be highlighted using a word cloud. Word clouds are widely used for analyzing data from social network websites. There is a Word Cloud image containing the words in the comments in the dataset used in Fig.3.

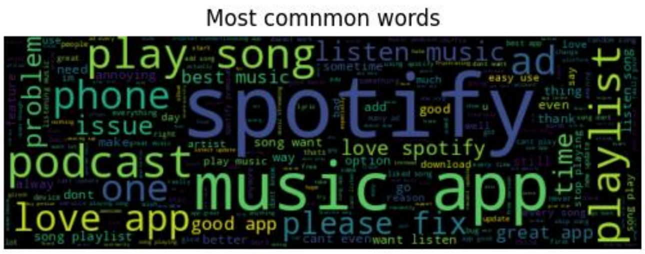

Fig. 3 Word Cloud of words in comments in dataset

D. *Feature Engineering*

In the Feature Engineering part, the splitting into train and test dataset, label encoding and text encoding are carried out. 67% of the dataset is reserved as train dataset and 33% as test dataset.

In the label encoding part, categorical data has been converted to numerical data. Thus, Negative class:0, Neutral class:1, Positive class:2.

In the text encoding part, 2 different methods were used. These are Count Vectorization and TF-IDF Character Level.

Count vectorization is a technique in natural language processing (NLP) for converting a piece of text into a numerical representation that can be used in machine learning algorithms [8].

The process involves counting the occurrences of each word in a document and creating a vector where each element represents the count of a particular word in the document. These vectors can then be used as input to various machine learning algorithms.

For example, consider the following sentence: "The quick brown fox jumped over the lazy dog." Using count vectorization, we would first create a vocabulary of unique words in the sentence, which would be: "the", "quick", "brown", "fox", "jumped", "over", "lazy", and "dog". We would then count the occurrence of each word in the sentence and create a vector representation of the sentence, which would be [2, 1, 1, 1, 1, 1, 1, 1], where each element in the vector corresponds to the count of the corresponding word in the vocabulary.

Count vectorization is a simple and effective way to represent text data for machine learning tasks such as classification and clustering.

TF-IDF (Term Frequency-Inverse Document Frequency) vectorization is a commonly used technique in natural language processing to convert a collection of textual documents into a numerical format that can be easily processed by machine learning algorithms [9].

TF-IDF vectorization involves two main steps:

(i) Term Frequency (TF) calculation: This step involves calculating the frequency of each term (word) in a document. The idea behind this step is that the importance of a term in a document is directly proportional to its frequency. The frequency of a term is calculated using the following formula:

$$TF(t,d) = \frac{\text{(Number of times term t appears in document d)}}{\text{(Total number of terms in document d)}} \qquad (1)$$

In equation (1), t refers to a term, d refers to a document, and TF(t, d) refers to the term frequency of term t in document d.

The ngram range value for TF-IDF was chosen as (2,3).

An n-gram is a contiguous sequence of n items from a given sequence of text, where an item can be a character, word, or any other unit of meaning.

For example, in the sentence "The quick brown fox", the 2-grams (or bigrams) would be "The quick", "quick brown", and "brown fox", while the 3-grams (or trigrams) would be "The quick brown", "quick brown fox".

N-grams can be used to model the structure and frequency of the sequence of text in a document or corpus. They are often used in text analysis and machine learning applications such as language modelling, sentiment analysis, and document classification.

The choice of n in an n-gram is a hyperparameter that can affect the quality of the resulting model. A higher value of n can capture longer sequences of text and potentially provide more context, but it can also result in more sparse data and higher computational costs. A lower value of n can result in more frequent and dense data but may not capture longer-term dependencies as well. The choice of the appropriate value of n depends on the specific task and the characteristics of the data.

E. *Machine Learning Classification*

After all preprocessing operations have been performed, a machine learning classification model has been created for the target column "Rating". When the data was examined before the model was created, it was seen that the distribution between Negative, Neutral and Positive classes was unbalanced. For this reason, the SMOTE algorithm was used first to make the data balanced. Synthetic Minority Oversampling Technique (SMOTE) is based on creating new minority values around the original values. Minority class examples are linked with their neighbours and new synthetic values are created with their linear combination, forming all new values in the minority class area [10].

The distribution of the classes before the SMOTE algorithm is used and the distribution of the classes after it is used are shown in Fig.4.

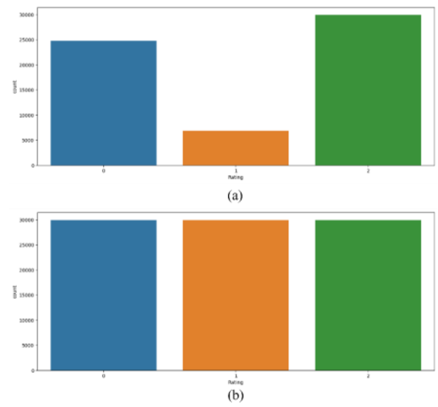

Fig. 4 (a) Before SMOTE processing (b) After SMOTE processing

After the SMOTE process, the number of elements of all classes became 29937. Thus, the imbalance in the dataset was eliminated. Then, a model was created using machine learning methods. Comparisons were made using the Logistic Regression, KNN, SVM, Decision Tree, Random Forest, XGBoost algorithms from machine learning methods. In the comparison, their performance was measured using Accuracy, Precision, Recall, F1-Score classification parameters.

In order to observe the effects of Text Encoding methods on success parameters, machine learning algorithms were first compared using the Count Vectorization method. The results are shown in Tab-1.

Table 1. Results of machine learning methods using Count Vectorization

Method	**Accuracy**	**Precision**	**Recall**	**F1-Score**
CV-LR	0.72	0.73	0.72	0.72
CV-KNN	0.47	0.61	0.47	0.44
CV-DT	0.59	0.60	0.60	0.60
CV-RF	0.70	0.71	0.70	0.70
CV-SVM	0.51	0.62	0.51	0.49
CV-XGBoost	**0.78**	**0.79**	**0.78**	**0.78**

The comparison, assessing the impact of employing Text Encoding methodologies, specifically utilizing TF-IDF, on machine learning approaches, is presented in Tab. 2.

Table 2. Results of machine learning methods using TF-IDF.

Method	**Accuracy**	**Precision**	**Recall**	**F1-Score**
TF-IDF-LR	0.68	0.68	0.68	0.68
TF-IDF-KNN	0.42	0.58	0.57	0.57
TF-IDF-DT	0.53	0.54	0.54	0.54
TF-IDF-RF	0.68	0.72	0.69	0.69
TF-IDF-SVM	0.49	0.49	0.49	0.49
TF-IDF-XGBoost	0.76	0.77	0.76	0.76

III. Results

When comparing Tab.1 and Tab.2, it becomes evident that the most successful approach, following the utilization of the Count Vectorization Text Encoding technique, is the subsequent application of the XGBoost algorithm. In the comparison between Count Vectorization and TF-IDF methods, it becomes apparent that the influence of the Count Vectorization technique on classification algorithms is notably more pronounced.

References

[1] Website. [Online]. Available: https://press.spotify.com/us/about/

[2] Jones, R. E. (2017) Keep it local: Music streaming and local music communities. (Master thesis, Miami University)

[3] Paradise, K. (2014) Digital music streaming in the 21st century: The music industry becomes radio-active. (Master thesis, Florida Atlantic University).

[4] Tusar, M. T. H. K., & Islam, M. T. (2021, September). A comparative study of sentiment analysis using NLP and different machine learning techniques on US airline Twitter data. In 2021 International Conference on Electronics, Communications and Information Technology (ICECIT) (pp. 1-4). IEEE.Retrieved from:https://arxiv.org/pdf/2110.00859.

[5] Sufi, F. K., & Khalil, I. (2022). Automated disaster monitoring from social media posts using AI-based location intelligence and sentiment analysis. IEEE Transactions on Computational Social Systems.

[6] Website. [Online]. Available: https://www.kaggle.com/datasets/mfaaris/spotify-app-reviews-2022

[7] Herbrich, R., Graepel, T., 2010. “Handbook of Natural Language Processing”, Chapman & Hall/CRC.

[8] Scikit-learn, Machine learning in python: [online]. [Accessed at 23 May 2020]. Retrieved from https://scikit-learn.org/stable/

[9] E. Cambria, B. Schuller, Y. Xia, and C. Havasi, “New Avenues in Opinion Mining and Sentiment Analysis,” Intelligent Systems, IEEE , vol.28, no.2, pp. 15-21, 2013.

[10] N. V. Chawla et al., “SMOTE: Synthetic Minority Over-sampling Technique”, Journal of Artificial Intelligence Research, vol. 16, pp. 321–357, Jun. 2002.