

Sentiment Analysis Using Naive Bayes Classifier

Ann Mary Ajith
PG Scholar

Master of Computer Application
Amal Jyothi College of Engineering
Kottayam, Kerala
annmaryajith2024a@mca.ajce.in

Gloriya Mathew
Assistant Professor

Master of Computer Application
Amal Jyothi College of Engineering
Kottayam, Kerala
gloriyamathew@amaljyothi.ac.in

Abstract— Sentiment analysis, a subfield of natural language processing, plays a vital role in understanding public opinion and sentiment towards products, services, or events. In this study, we explore the field of sentiment analysis with a special emphasis on the use of machine learning techniques to classify the sentiments set in textual data. A Multinomial Naive Bayes classifier trained on a dataset of text data with sentiment markers is used in the study.

The text data is pre-processed and vectorized using the Term Frequency-Inverse Document Frequency (TF-IDF) technique. This helps with a variety of operations, including request exploration, client feedback analysis, and social media monitoring. The study looks at the techniques used, assesses the effectiveness of the model, and addresses about the results and possible directions for farther sentiment analysis exploration.

Keywords— Sentiment Analysis, Natural Language Processing, Machine Learning, Multinomial Naive Bayes, TF-IDF, Classification, Opinion Mining

I. INTRODUCTION

Sentiment analysis, occasionally appertained to as opinion mining, is the computational analysis of opinions, attitudes, and emotions expressed in textual data. As user-generated information on blogs, social media platforms, and review websites grows at an exponential rate, sentiment analysis is getting further and further vital for investigators, marketers, and companies to understand how the public feels about certain goods, services, or events. Having the capacity to automatically estimate and classify textual data according to sentiment opposition (positive, negative, or neutral) helps businesses improve client satisfaction, make well-informed opinions, and customize their marketing campaigns.

Machine learning techniques have come largely effective tools for sentiment analysis tasks in recent times. Among these techniques, the Naive Bayes classifier is particularly notable for text type tasks due to its ease of use, efficiency, and effectiveness. Naive Bayes classifiers are probabilistic models that calculate on the independence of features and the Bayes theorem. Naive Bayes classifiers, despite their simplicity, have proven to perform competitively in a variety of text type operations, similar as sentiment analysis. The Naive Bayes classifier is used in this paper's exploration study on sentiment analysis. The purpose of the study is to examine how well textual data is used to predict sentiment polarity using the Naive Bayes classifier. We make use of a

dataset that includes tagged text samples that indicate two different moods negative and positive.

Preprocessing is applied to the text data in order to count noise, normalize the text, and extract relevant features. The textual data is also vectorized using the Term frequency-Inverse Document frequency (TF-IDF) technique to produce numerical point vectors that may be used with machine learning algorithms. Then, using the vectorized data, we train a Multinomial Naive Bayes classifier to predict sentiment polarity.

II. RELATED WORK

The study in [1] aimed to dissect text documents from Twitter concerning public policies related to COVID-19 using Multinomial Naive Bayes for sentiment classification. The exploration followed the CRISP-DM methodology, covering business understanding, data understanding, data preparation, modelling, and evaluation stages. Multinomial Naive Bayes was employed for text type. The findings illuminate the effectiveness of the model in classifying text sentiments, achieving a high accuracy of 90.25%. Specifically, regarding the new normal policy, over 70% of social media users expressed support. The novelty of this exploration lies in demonstrating the successful operation of Multinomial Naive Bayes for sentiment analysis on COVID-19-related public policies, slipping light on global opinions and attitudes towards the new normal policy.

The paper [2] introduces a movie recommendation system exercising Cosine Similarity to suggest analogous cinema rested on user preferences. In addition to traditional recommendation styles, the system incorporates sentiment analysis of movie reviews using Naive Bayes (NB) and Support Vector Machine (SVM) classifiers to assess whether a movie is worth watching. A relative analysis between NB and SVM rested on criteria similar as Accuracy, Precision, Recall, and F1 Score reveals SVM's superiority, achieving a accuracy score of 98.63 compared to NB's 97.33. This highlights SVM's effectiveness in sentiment analysis and its implicit to enhance user experience in movie recommendation systems.

The study in [3] delves into the intricate task of decoding sentiments and emotions from text, admitting its complexity compared to other modalities like facial expressions or voice. Emotion recognition from text has garnered significant interest among experimenters due to its wide-ranging

operations, including recommendation systems, mood tracking, and multimedia tagging. With the rise of information-rich surroundings, the significance of smart sociotechnical systems suitable of analyzing data for perceptivity has come evident. Recent advancements in AI, particularly in health monitoring and communication technologies, have eased sentiment identification. The study focuses on seven emotional countries and employs a Correlation- rested Naive Bayes Classifier to dissect user textbook emotions on social media platforms, achieving an emotional accuracy rate of 99.99%.

The research substantiated in [4] leverages the RStudio application and the Naive Bayes Classifier Algorithm to analyze both positive and negative reviews of an e-commerce operation from user reviews on Google Play. The technique employed is text mining and text processing, involving way similar as stemming, tokenizing, case folding, normalization, and filtering. Review data for the Shopee app on Google Play is collected through web scraping using the appfollow web operation, and the data is saved in CSV format. The acquired data undergoes text processing, starting with rephrasing reviews in other languages to Indonesian and also homogenizing the content to remove emoticons. Next, the homogenized data is converted to lowercase letters in the case folding process. Each word that's independent, or has no influence, is isolated as a token. Tokenizing facilitates the computation of word presence and frequency in the data. The Naive Bayes Classifier Method is employed to classify sentiments as positive or negative based on word frequency after text processing.

The study in [5] delves into the critical role of education as a pivotal indicator of progress and well-being, aligning with the UN's Sustainable Development Goal 4 (SDG4) aimed at ensuring inclusive and equitable quality education for all. With India's adoption of this agenda in 2015, there's an increased focus on the need for educational institutions to closely monitor student academic performance to identify and support low-performing students effectively. To address this, the study proposes leveraging sentiment analysis on social media platforms such as Twitter and Facebook to gauge students' opinions and feelings towards their educational institution, thus providing insights into quality education indicators. Conducting a systematic review of 21 studies indexed in SCOPUS, the research focuses on sentiment analysis related to SDG4 quality education. The dataset utilized for analysis, sourced from Kaggle, comprises two files representing student performance in Math and Portuguese language courses. Through visualization and analysis, the study concludes that the Support Vector Machine (SVM) model stands out as stable and effective, particularly in handling nonlinear data using Kernel techniques.

The paper [6] addresses the challenges in sentiment analysis of social networking service (SNS) data using the Naive Bayes algorithm. Existing methods often employ the same number of attributes to estimate the weight of each

class, leading to decreased accuracy due to the inclusion of numerous uncountable and meaningless attributes. To overcome these issues, the paper proposes two methods. Firstly, it suggests adjusting the calculation of weights based on the difference between the number of positive and negative words. Secondly, it introduces a feature selection step using the Multinomial Naive Bayes (MNB) algorithm to eliminate insignificant words. Performance comparison reveals that the proposed scheme significantly improves accuracy compared to existing methods such as the Multivariate Bernoulli Naive Bayes (BNB) algorithm and MNB scheme.

III. METHODOLOGY

The methodology section describes the process followed to conduct the research study, including data collection, preprocessing, feature extraction, classifier training, and performance evaluation.

A. Data Collection:

The research utilizes a dataset consisting of labeled text samples representing different sentiments, including positive, negative, and neutral. The dataset is collected from publicly available sources such as social media platforms, review websites, and sentiment analysis repositories.

B. Data Preprocessing:

Before training the classifier, the text data undergoes preprocessing to remove noise, standardize text format, and extract relevant features. Preprocessing steps include tokenization, stop word removal, punctuation removal, and stemming or lemmatization.

C. Feature Extraction:

To convert the textual data into numerical feature vectors suitable for machine learning algorithms, the Term Frequency-Inverse Document Frequency (TF-IDF) technique is employed. TF-IDF assigns weights to words based on their frequency in a document relative to their frequency across all documents in the corpus.

D. Classifier Training:

A Multinomial Naive Bayes classifier is trained on the vectorized data to predict sentiment polarity. Naive Bayes classifiers are probabilistic models that calculate the probability of a document belonging to a particular class based on the presence of features.

E. Performance Evaluation:

The performance of the classifier is evaluated using metrics such as accuracy, precision, recall, and F1-score. Additionally, a classification report and confusion matrix are generated to assess the classifier's performance across different sentiment classes.

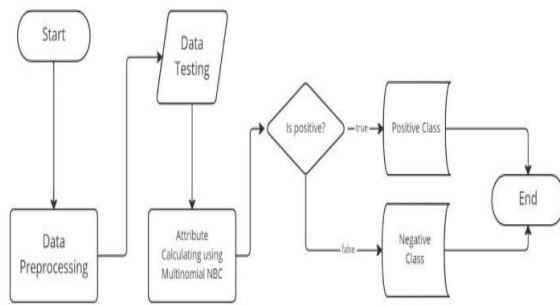


Fig 1. Flowchart

```

train_data = pd.read_csv('train.csv')
X_train = train_data['text']
y_train = train_data['sentiment']

test_data = pd.read_csv('test.csv')
X_test = test_data['text']
y_test = test_data['sentiment']
  
```

Fig 3. Loading dataset

F. Implementation

1. Environment Setup

Begin by setting up the development environment. Ensure you have Python installed along with necessary libraries such as tkinter, scikit-learn, and pandas.

2. Code Structure

Create a Python script named sentiment_analysis.py or any preferred name. Import the required libraries and modules, including tkinter, scrolled text, and the necessary modules from sklearn. Load the training and testing data from CSV files using Pandas. Vectorize the data using the TF-IDF technique. Train the Naive Bayes classifier and evaluate its accuracy.

```

import tkinter as tk
from tkinter import scrolledtext
from tkinter import font
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import classification_report, confusion_matrix
import pandas as pd
import re
  
```

Fig 2. Importing libraries and modules

3. Load and Preprocess Data:

Here, you load the training and testing data from CSV files into Pandas DataFrames. After loading the data, you preprocess the text data by removing any special characters, converting text to lowercase, and performing other necessary preprocessing steps.

4. Model Training:

In this step, you vectorize the preprocessed text data using the TF-IDF (Term Frequency-Inverse Document Frequency) technique. This converts the textual data into numerical feature vectors suitable for machine learning algorithms. Then, you train the Naive Bayes classifier using the vectorized training data.

```

vectorizer = TfidfVectorizer()
X_train_vectorized = vectorizer.fit_transform(X_train)
X_test_vectorized = vectorizer.transform(X_test)

nb_classifier = MultinomialNB()
nb_classifier.fit(X_train_vectorized, y_train)

accuracy = nb_classifier.score(X_test_vectorized, y_test)
print(f"Model Accuracy: {accuracy}")
  
```

Fig 4. Training model

5. GUI Design and User Input Handling:

This step involves designing the graphical user interface (GUI) using Tkinter. You create components such as a text area for user input, a button to trigger sentiment analysis, and a label to display the predicted sentiment. Additionally, you implement functions to handle user input and trigger sentiment analysis, ensuring that the GUI remains responsive during prediction.

```
def handle_user_input():
    new_text = text_area.get(index="1.0", index2="end-1c").strip()
    if not new_text:
        result_label.config(text="Please enter valid text.")
        return
    if re.match(pattern="[a-zA-Zs]+$", new_text):
        result_label.config(text="Input should contain at least one alphabetic character.")
        return
    root.after(100, predict_sentiment_async, new_text)

root = tk.Tk()
root.title("Sentiment Analysis")

title_font = font.Font(family="Helvetica", size=16, weight="bold")
button_font = font.Font(family="Helvetica", size=12)
label_font = font.Font(family="Helvetica", size=12)

text_area = scrolledtext.ScrolledText(root, width=50, height=5, font=label_font)
text_area.pack(pady=10)

analyze_button = tk.Button(root, text="Analyze", command=handle_user_input, font=button_font)
analyze_button.pack(pady=5)

result_label = tk.Label(root, text="", font=label_font)
result_label.pack(pady=10)

root.mainloop()
```

Fig 5. GUI implementation

IV. RESULT

This section presents the experimental results obtained from training and evaluating the Naive Bayes classifier for sentiment analysis. The results include accuracy scores, classification reports, and confusion matrices, providing insights into the classifier's performance across different sentiment classes.

```
Model Accuracy: 0.82952
Classification Report:
              precision    recall  f1-score   support

   Negative      0.79      0.89      0.84     12500
    Positive      0.87      0.77      0.82     12500

 accuracy              0.83     25000
 macro avg      0.83      0.83      0.83     25000
weighted avg      0.83      0.83      0.83     25000

Confusion Matrix:
[[11112 1388]
 [ 2874 9626]]
```

Fig 6. Result

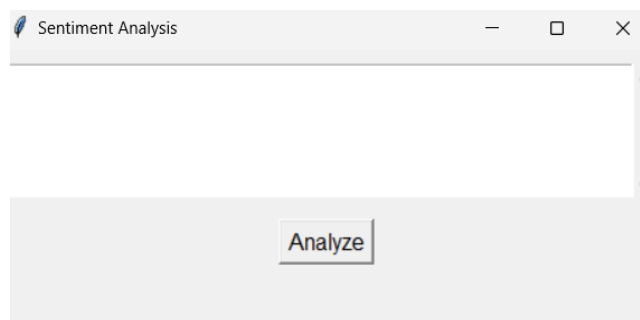


Fig 7. Output of user interface

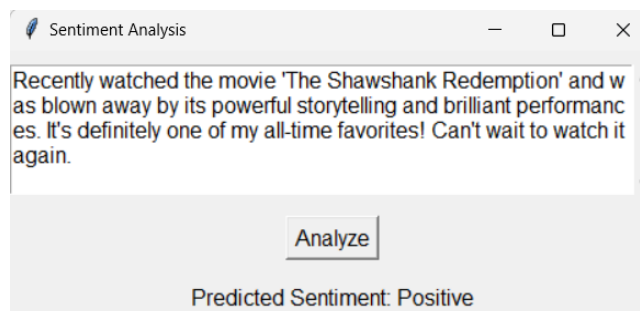


Fig 8. Output of result

V. CONCLUSION

An expansive study of sentiment analysis with the Naive Bayes classifier is presented in this publication. The study shows how well the classifier predicts sentiment polarity from textual data, pressing its possibility for practical uses in a range of domains like social media monitoring, client support, and marketing. The findings of the experimental highlight how essential it is to understand sentiment in text data. Organizations can gain important insight into public opinion, consumer feedback, and request trends by precisely grading sentiments as positive or negative. Businesses can improve consumer satisfaction, customize marketing strategies, and make data-driven opinions.

Sentiment analysis's capacity to develop useful perceptivity from enormous volumes of unstructured textual data is one of its main features. Businesses may efficiently handle and analyze massive quantities of client reviews, social media posts, and online debates by automating the sentiment classification process. This helps businesses to remain flexible and adaptable in the fast-paced digital world while also saving time and resources.

Sentiment analysis is a useful tool for perfecting user experiences and can be integrated into a range of platforms and applications. Recommendation systems, chatbots, and virtual assistants can give further applicable and intriguing content, goods, and services by evaluating user sentiments. This builds better bonds with users and increases their loyalty.

Overall, this study's findings demonstrate the significance of sentiment analysis as a useful tool for gaining insight into

client sentiment, perfecting decision-making, and fostering commercial success. Sentiment analysis will come more and more important in impacting marketing strategies, enhancing client experiences, and promoting invention in the digital period as businesses continue to borrow data-driven ways.

VI. REFERENCE

- [1] ZULFIKAR, Budiawan W., ATMADJA, Rialdy A. and PRATAMA, Fajri S., "Sentiment analysis on social media against public policy using multinomial naive bayes", Scientific Journal of Informatics, 10 (1), 25-34,2023.
- [2] N Pavitha, Vithika Pungliya, Ankur Raut, Roshita Bhonsle, Atharva Purohit, Aayushi Patel, R Shashidhar, "Movie recommendation and sentiment analysis using machine learning", Global Transitions Proceedings, Volume 3, Issue 1, 2022.
- [3] Akhilesh Kumar and Awadhesh Kumar, "Human Sentiment Analysis on Social Media through Naïve Bayes Classifier", Journal of Scientific Research, Volume 66, Issue 1, 2022.
- [4] Saputri Y. R., Februariyant H., "Sentiment analysis on shopee e-commerce using the Naive Bayes classifier algorithm", Jurnal Mantik, 6(2) ,2022.
- [5] Pooja, Bhalla, R. "A Review Paper on the Role of Sentiment Analysis in Quality Education.", SN COMPUT. SCI. 3, 469 ,2022.
- [6] J. Song, K. T. Kim, B. Lee, S. Kim, H. Y. Youn, "A novel classification approach based on Naïve Bayes for Twitter sentiment analysis," KSII Transactions on Internet and Information Systems, vol. 11, no. 6, pp. 2996-3011, 2017.