AI Model for Sentiment Analysis System Using Python

DR. SANTOSH SINGH[1], RIMSY DUA[2], KIRAN PAL[3], LISA RODRIGUES[4]
[1] *Head of Department, Department of IT, Thakur College of Science and Commerce, Thakur Village, Kandivali (East), Mumbai, Maharashtra, India*
[2]*Assistant Professor, Department of IT, Thakur College of Science and Commerce, Thakur Village, Kandivali (East), Mumbai, Maharashtra, India*
[3,4] *PG Student, Department of IT, Thakur College of Science and Commerce, Thakur Village, Kandivali (East), Mumbai, Maharashtra, India*

Abstract— Sentiment analysis is an important task in natural language processing that aims to understand and classify opinions and emotions represented in text data. This research article introduces a complete way to develop an AI model for sentiment analysis using the Python programming language. The study makes use of a labeled Twitter dataset, which includes messages with positive, negative, and neutral attitudes. On this dataset, machine learning model is trained and assessed, with a focus on preprocessing stages such as text cleaning, tokenization, and feature extraction. The suggested system makes use of cutting-edge machine learning and deep learning algorithms, as well as Natural Language Processing (NLP) libraries. Data pre-processing, feature extraction, and the use of various text representation approaches, such as word embeddings, are all part of the model construction process.

Indexed Terms— Random Forest, Health, Text, Machine Learning, Sentiment, Voice, Twitter

I. INTRODUCTION

Social media platforms like Twitter have become important sources for evaluating public mood in the digital age when information is continually flowing through multiple online channels. Social media platforms allow the public to express their emotions[1]. Businesses, researchers, and individuals can all benefit from analyzing the mood expressed in tweets. Sentiment analysis, often known as opinion mining, is an important subfield of natural language processing (NLP) that seeks to assess and categorize the emotions and opinions expressed in text data. Sentiment analysis of tweets is a challenging task due to short tweet length, frequent informal use, and rapid language evolution in Twitter [5].. With advancements in AI and Python, there is a chance to create a system that can assist with sentiment prediction. This research delves into sentiment analysis by using machine learning algorithms to classify tweets into three categories: positive, negative, and neutral. The proposed project is also beneficial to disabled people who are blind or physically disabled and cannot type or write. The previous paper lacked a voice assistant-based system that could assist disabled people, so this paper fills that void. Text-based depression detection systems are gaining popularity because they offer a scalable and non-intrusive method of assessing mental health. Python is a popular programming language that includes many libraries and tools that can be used to create an AI application for sentiment analysis systems. Python also has a very rich machine-learning ecosystem, making it the best choice for developing the sentiment analysis system. To summarize, the sentiment analysis system built with Python and AI offers a promising approach with a wide range of practical applications, such as tracking real-time social media sentiment trends, customer feedback analysis, and voice-activated sentiment analysis tools. It is a valuable resource for those looking to use sentiment analysis to make data-driven decisions and understand public opinion. The sections that follow provide a detailed look at each component of this comprehensive system.

II. LITERATURE REVIEW

The research paper of Rijwan Khan, Piyush Shrivastava, Aashna Kapoor, Aditi Tiwari, Abhyudaya Mittal dealt with AI based Social Media Analysis. They proposed a sentiment analysis system

for Twitter, analyzing people's reactions to government and local authorities' decisions. The system uses automation and natural language processing to categorize tweets into positive, negative, or neutral sets, achieving accuracy, quantization, and prediction [1].

The researchers BiswaRanjan Samal; Anil Kumar Behera; Mrutyunjaya Panda have done analysis of the performance of supervised machine learning techniques for sentiment analysis. This paper collects movie review datasets and selects popular supervised machine learning algorithms for training a model to categorize reviews. Python's NLTK package, WinPython, and Spyder are used for processing, while Python's sklearn package is used for training and accuracy [2].

In Machine Learning Sentiment Analysis of Yelp Reviews by Hemalatha S.; Ramathmika Ramathmika, analysed Yelp reviews to determine their positive or negative sentiment. The study uses ma- chine learning algorithms from the nltk library of Python, comparing their efficiency based on their effectiveness in Natural Language Processing research [3].

The authors [4] presented an approach based on KNN classifiers for multi-class sentiment analysis of Twitter data. This study uses Twitter data for sentiment classification using a KNN classification algorithm, an improvement over an existing SVM method. The data was extracted using Python's Tweepy, and the supervised machine learning algorithm k-nearest neighbor was used for sentiment classification. The proposed technique outperforms the existing method in accuracy, precision, and recall.

III. ALGORITHM

Random forest, a popular ensemble machine learning algorithm, is used for classification tasks in this paper. The algorithm analyses sentiment using a dataset of text data. This algorithm divides the dataset into training and testing sets, builds a random forest classifier, trains it on the training data, and assesses its performance using accuracy and classification reports. There are three distinct datasets: positive, negative and neutral. These files contain text data and labels that indicate whether the sentiment is positive, negative or neutral. The data is divided into 80% training and 20% testing sets. The Random Forest ensemble machine learning algorithm is widely used for classification and regression tasks. It is a decision tree ensemble in which multiple decision trees are combined to produce more accurate and reliable results.

Sentiment analysis is a critical natural language processing task that involves determining the sentiment or emotional tone expressed in text data, such as whether it is positive, negative, or neutral. The Random Forest algorithm, a powerful ensemble learning method, is used in this study to classify text data from Twitter into these sentiment categories.

One of the most important steps in sentiment analysis is converting raw text data into a numerical format that machine learning models can use. The Term Frequency-Inverse Document Frequency (TF-IDF) vectorization method is used in this study. It converts text data into numerical features that represent the significance of each word in the document about the entire dataset.

A Random Forest classifier is used to classify sentiment. To make predictions, this ensemble learn- ing algorithm employs multiple decision trees. It is started and trained with training data. For model training, the TF-IDF transformed text data (X_train_vectorized) and sentiment labels (y_train) are used. The Random Forest algorithm, which serves as the core machine learning model for sentiment analysis, is a critical component of this research. It employs ensemble techniques to make accurate predictions about sentiment, and its effectiveness is measured using the accuracy metric.

IV. DATASET

Datasets: The Twitter dataset was used as the source for the text in the study. There are 1000 text data in total. The dataset is divided into three sections: positive, negative and neutral text. The in- formation is in JSON format. The "text" field from each JSON file is extracted first, and the extracted texts are saved in a Pandas Data Frame, which is then saved to an Excel file for further processing or analysis. Regular expressions and the re-library are used to power the

text-cleaning procedures. It performs the following actions:

- Removes special characters and emojis by replacing them with an empty string.
- Removes hyperlinks by replacing them with an empty string.

V. METHODOLOGY

Text classification:

To perform sentiment analysis, the Python program employs a Random Forest classifier with TF-IDF (Term Frequency-Inverse Document Frequency) vectori- zation. The purpose of this code is to create a sentiment analysis model that can predict whether a given text has a positive, negative or neutral sentiment and then test its accuracy on a test dataset. The model is built using the Random Forest algorithm and TF-IDF features extracted from text data. The code reads three Excel files, one with positive sentiment data and the others with negative and neutral sentiment data, and stores them in separate Pandas Data Frames. This dataset combines both positive, negative and neutral sentiment data for training and testing. The text data is vectorized using TF-IDF. This procedure converts text into numerical feature vectors that can be fed into machine learning models. Finally, the code prints the sentiment analysis model's accuracy on the test data.

Sentiment Prediction:

Speech Recognition- It uses the computer's microphone to capture audio, adjusts for ambient noise, and records the speech input. It uses Google Web Speech API to recognize the speech input and print the recognized text. For sentiment analysis, it appears to assume a previously defined TF-IDF vectorizer and a random forest classifier. Using the same TF-IDF vectorizer, it vectorizes the recognized sentence. If the predicted sentiment is 4, it displays "Positive sentiment", if the predicted sentiment is 0, it displays "Negative sentiment"; otherwise, it displays "Neutral sentiment".

Consider the following:

- Make sure you've imported the necessary libraries (speech_recognition and any other libraries needed for sentiment analysis).
- Before running this code, ensure that the vectorizer and random_forest variables have been properly defined and trained.
- Ascertain that your computer's microphone is configured and ready for audio input.
- This code essentially listens to what you say, transcribes it, and then analyses the transcribed text for sentiment. It combines speech recognition with text-based sentiment analysis

Keywords Extraction- The code provided is a Python script that performs several tasks related to analyzing and visualizing text data from two Excel files containing positive, negative and neutral sentiment data.

Word Cloud Visualisation:

Word clouds are a visual representation of frequently occurring words that can aid in the identification of common themes or topics in text data. This code reads sentiment data from three Excel files, pre-processes it, explores it, and visualizes it. It uses a word cloud to provide insights into the distribution of sentiment labels and the most common words in text data.

VI. RESULT

The proposed project analyses a word or a sentence to determine whether it expresses a positive, negative or neutral sentiment and then provides a result based on that analysis. The system's voice assistant will assist disabled users (e.g., users who cannot use the keyboard/mouse or users who are blind). Random forest algorithm was used that gave an accuracy of 85%. The dataset was obtained from the Kaggle website.

Sentiment Analysis Result: After training the Random Forest Classifier on your dataset, the code predicts sentiment labels for the test data (either 'Positive', ‘Negative’ or 'Neutral'). The sentiment classification accuracy is calculated and printed.

Speech Recognition Result: The code captures audio from the microphone, attempts to recognize the spoken words using Google Web Speech API, and prints the recognized text. If the speech is clear and successfully recognized, the recognized text will be displayed as "You said: [text]." It will print if the

speech is unclear or not recognized. "Sorry, I could not understand what you said."

Sentiment Prediction from Spoken Text: The recognized speech text is vectorized using the same TF-IDF vectorizer that was used for training. The sentiment of the spoken sentence is predicted by the trained Random Forest Classifier. The predicted sentiment is printed ('Positive sentiment', 'Negative sentiment' or 'Neutral sentiment ').

The clarity of your speech and the quality of the microphone are critical to the success of the speech recognition section. The sentiment analysis is dependent on the quality of the training data and the Random Forest Classifier's efficacy.

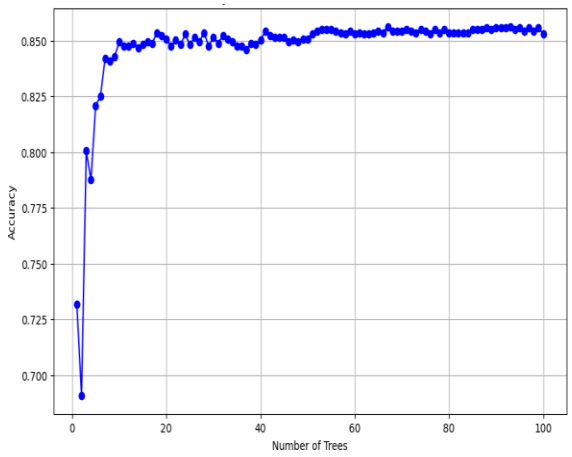

Figure 1. Accuracy vs. Number of Trees in Random Forest

CONCLUSION

This research project demonstrated the development and evaluation of a robust sentiment analysis system in the field of natural language processing and sentiment analysis. The system presents a comprehensive solution for understanding and classifying sentiments expressed in text data, with a focus on Twitter content, by leveraging cutting-edge machine learning techniques and innovative integration with a voice assistant. The use of machine learning algorithm Random Forest, demonstrates the ability to classify senti- ments as positive, negative or neutral. By using this algorithm, businesses and individuals can gain insights from social media content, customer feedback, and other sources. Furthermore, this study demonstrated the system's ability to predict sentiments not only from text but also from spoken words, bridging the gap between textual and verbal communication. The sentiment analysis system achieves impressive results in categorizing text data into positive, negative, and neutral sentiments by utilizing Random Forest machine learning algorithm. The addition of a voice assistant improves the system's usability and accessibility. The system's adaptability makes it suitable for a wide range of applications. It can be used to track social media sentiment trends, monitor customer feedback, and provide quick sentiment analysis via voice recognition, providing individuals, businesses, and researchers with valuable insights. This research project not only contributed to the field by developing a robust and accurate sentiment analysis system, but it also broadened its potential use cases by incorporating voice recognition capabilities. Finally, the sentiment analysis system presented in this paper makes an important contribution to the field of natural language processing. It emphasizes the significance of understanding sentiment in both written and spoken communication and provides a user-friendly tool with numerous applications.

REFERENCES

[1] Khan, Rijwan, et al. "Social media analysis with AI: sentiment analysis techniques for the analysis of Twitter covid-19 data." J. Crit. Rev 7.9 (2020).

[2] Samal, BiswaRanjan, Anil Kumar Behera, and Mrutyunjaya Panda. "Performance analysis of supervised machine learning techniques for sentiment analysis." 2017 Third International Conference on Sensing, Signal Processing and Security (ICSSS). IEEE, 2017.

[3] Hemalatha, S., and Ramathmika Ramathmika. "Sentiment analysis of Yelp reviews by machine learning." 2019 International Conference on Intelligent Computing and Control Systems (ICCS). IEEE, 2019.

[4] Hota, Soudamini, and Sudhir Pathak. "KNN classifier-based approach for multi-class sentiment analysis of Twitter data." Int. J. Eng. Technol 7.3 (2018): 1372-1375.

[5] T. Carpenter, and T. Way, "Tracking Sentiment Analysis through Twitter,". ACM computer survey. Villanova: Villanova University, 2010.

[6] Saif, H., He, Y., Alani, H. (2012). Semantic Sentiment Analysis of Twitter. In: Cudré-Mauroux, P., et al. The Semantic Web – ISWC 2012. ISWC 2012.

[7] Sarlan, Aliza & Nadam, Chayanit & Basri, Shuib. (2014). Twitter sentiment analysis. 212-216.

[8] Kharde, Vishal & Sonawane, Sheetal. (2016). Sentiment Analysis of Twitter Data: A Survey of Techniques. International Journal of Computer Applications. 139. 5-15.

[9] M. Hagen, M. Potthast, M. Büchner, and B. Stein, "Webis: An ensemble for Twitter sentiment detection," in Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015), 2015, pp. 582-589.

[10] Sarkar, Dipanjan. Text analytics with Python: a practitioner's guide to natural language processing. Bangalore: Apress, 2019.

[11] Neelakandan, S., and D. Paulraj. "An automated learning model of conventional neural network-based sentiment analysis on Twitter data." Journal of Computational and Theoretical Nanoscience 17.5 (2020)

[12] Razno, Maria. "Machine learning text classification model with NLP approach." Computational Linguistics and Intelligent Systems 2 (2019): 71-73.