

## Twitter Sentiment Analysis using Python (TweetX)

Vijayalaxmi Tadkal<sup>1</sup>, Ajit Yadav<sup>2</sup>, Amit Ghaytadake<sup>3</sup>, Vivek Salwalkar<sup>4</sup>, Mahesh Yadav<sup>5</sup>

<sup>1</sup>Asst Prof Department of Computer Science Engineering [AIML] & IETE'S Bharat College of Engineering

<sup>2</sup>BE Department of Computer Science Engineering [AIML] & IETE'S Bharat College of Engineering

<sup>3</sup>BE Department of Computer Science Engineering [AIML] & IETE'S Bharat College of Engineering

<sup>4</sup>BE Department of Computer Science Engineering [AIML] & IETE'S Bharat College of Engineering

<sup>5</sup>BE Department of Computer Science Engineering [AIML] & IETE'S Bharat College of Engineering

**Abstract** - Twitter has become a significant platform for expressing opinions and emotions on a wide range of topics. This study presents a comprehensive analysis of sentiment expressed in Twitter data using natural language processing and machine learning techniques. The primary objective is to gauge public sentiment on various subjects, from politics to brand perception, by analysing tweets. We collected a vast dataset of tweets, pre-processed the text, and employed sentiment analysis algorithms to classify tweets as positive, negative, or neutral. The results provide valuable insights into public sentiment trends, allowing businesses, policymakers, and researchers to make informed decisions based on the collective voice of Twitter users. The study demonstrates the potential of sentiment analysis as a tool for understanding public sentiment and monitoring real-time public opinion

**Key Words:** (NLTK), TextBlob

### 1. INTRODUCTION

For a human mind it is very easily too thin of new content. what if someone asks you to “draw a flower with blue petals”. It is very easy for us to do that. but machines process information very differently. Just understanding the structure of the above sentence is a difficult task for them let alone generate something based on that description. Automatic synthetic content generation is a field that has been explored in the past and was discredited because at that time neither the algorithms existed nor enough processing power that could help solve the problem. However, the advent of deep learning started changing the earlier beliefs. The tremendous power of neural networks to capture the features even in the humongous of datasets makes them a very viable candidate for automatic content generation. another milestone milestone was achieved when Ian Good Fellow proposed generative adversarial networks in 2014. GANs are a kind of architecture in Deep learning that can produce content from random noise. What is even more unique about GANs is that the content they create represents the dataset on which they are being trained upon but it is totally unique in some way or the other. Generating an image from a text-based description is one aspect of generative adversarial networks that we will focus upon. Since the GANs follow unsupervised learning approach we have modified them to take an input as a condition and generate based on the input condition. This can form base for a large number of things like synthetic audio generation like the ones used in Siri or assistant, video content generation from just scripts. imagine entire movies made out of just the script. These are some uses that many companies are

researching about. . modifying GANs and applying conditions on them isn't limited to just generating images, we can use it to create passwords that are very hard to crack and numerous similar applications like this.

### 2. EXISTING SYSTEM

PyTorch is a popular Py lib for d learning and MLearning . PyTorch can be used to build a variety of image generators, including (GAN) and (VAEs).

TensorFlow is another popular Python library for deep learning and ML. TensorFlow can also be used to build a variety of image generators, including GANs and VAEs.

Jax is a relatively new Python library for mL learning and dL learning. Jax is gaining popularity due to its ease of use and flexibility. Jax can also be used to build a variety of image generators.

### 3. LITERATURE REVIEW

To understand how to generate content required an extensive study of Deep Learning and Unsupervised learning and to do that we used various Books and Publication along with Some Blogs, Talks and Conferences. Pre-process the words into embedding. Taking in input as a word and do proper tokenization. Creating an LSTM Model and generating embedding using the model. Using a LSTM model, we need to convert the long sentence description into a word embedding to pass it into the generator □ Pre-processing the image and using proper techniques to normalize them. Normalizing and augmenting the images using own methods in jumpy.

1) Natural Language Processing with Python by Steven Bird

In this book I got familiarized with a lot of tools and techniques to process words and the Overall field of Natural Language Processing and the best practices to properly process my Input

2) Neural Networks and Deep Learning by Michael Nielsen

In this book I studied various architectures of Deep Learning that are most commonly used today along with proper techniques to train the models and steps to avoid overfitting and underfitting .

3) Generative Adversarial Networks by Ian Goodfellow

Ian Goodfellow is the creator of Generative Adversarial Networks and this publication helped me understand what GANs are, how they function and more importantly how can they even do what they do. Also Introduced me to some Problems like Mode Collapse.

#### 4. METHODOLOGY

HTML, is the commonly used language for creating and structuring content on the site ,HTML helps organize information through elements like headings, paragraphs, and lists. It allows for the inclusion of hyperlinks to navigate between sections or external sources. HTML also supports multimedia integration, ensures accessibility, aids discoverability through metadata, and enables interactivity for engaging readers. Overall, HTML serves as a fundamental tool for presenting research effectively online.

CSS is a sheet-style language used to define the appearance and format of a markup document. It gives HTML a supplementary function. Used with HTML, the style of user interfaces and webpage is changed. It may also be used in XML documents of any form, including simple XUL, SVG and XML documents. In most websites, CSS is used with HTML and JavaScript to develop web-based user interfaces and user interfaces for a variety of mobile applications. What CSS accomplishes is: you can add unique appearances to your HTML pages, modify the style with just a few modifications to CSS code. C.S.S. is used in the creation of HTML Tags. C.S.S. is used widely used as a web language, to create a web page, we usually use H.T.M.L., C.S.S. and JavaScript. CSS is also a commonly used language in Cascading Style sheet. It allows web developers to use HTML tags for styling.

JavaScript or JS is an object oriented light weight language used for web page scripting by various online sites. The HTML document is a fully interpreted computer language allowing interactivity dynamically on web pages. In 1995, it was launched to add software to Netscape Navigator's web pages. All other graphical web browsers have been embraced since then. Users may construct contemporary web applications with JavaScript so that they can interact without refreshing the page at all times. Js is used in the conventional website for various sorts of easiness and interaction

#### Backend Technologies

Python is a commonly used programming language celebrated for its simplicity, readability, and adaptability across various domains. Its intuitive syntax and dynamic typing make it easy to learn and use, appealing to beginners and experienced developers alike. Python's interpreted nature allows for interactive testing and debugging, fostering rapid development cycles. With a vast common library and a thriving ecosystem of third-party packages, Python is well-suited for diverse applications, Its cross-platform compatibility ensures seamless execution on different operating systems, while its strong community support and governance by the Python Software Foundation ensure continuous growth and evolution. Overall, Python's versatility and extensive resources developers tackling a big range of projects.

#### 5. Architecture & Data flow diagram

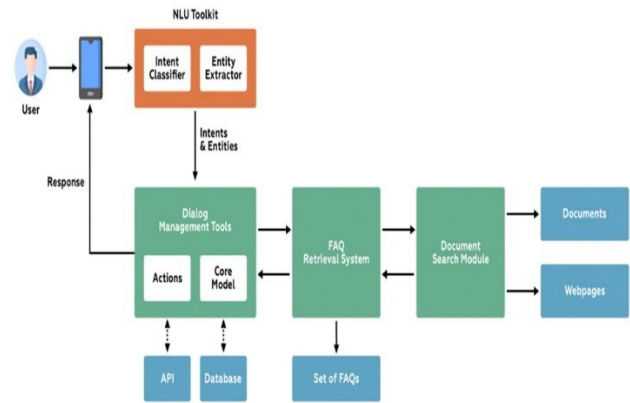


Fig 1: Architecture

#### Data Flow Diagram

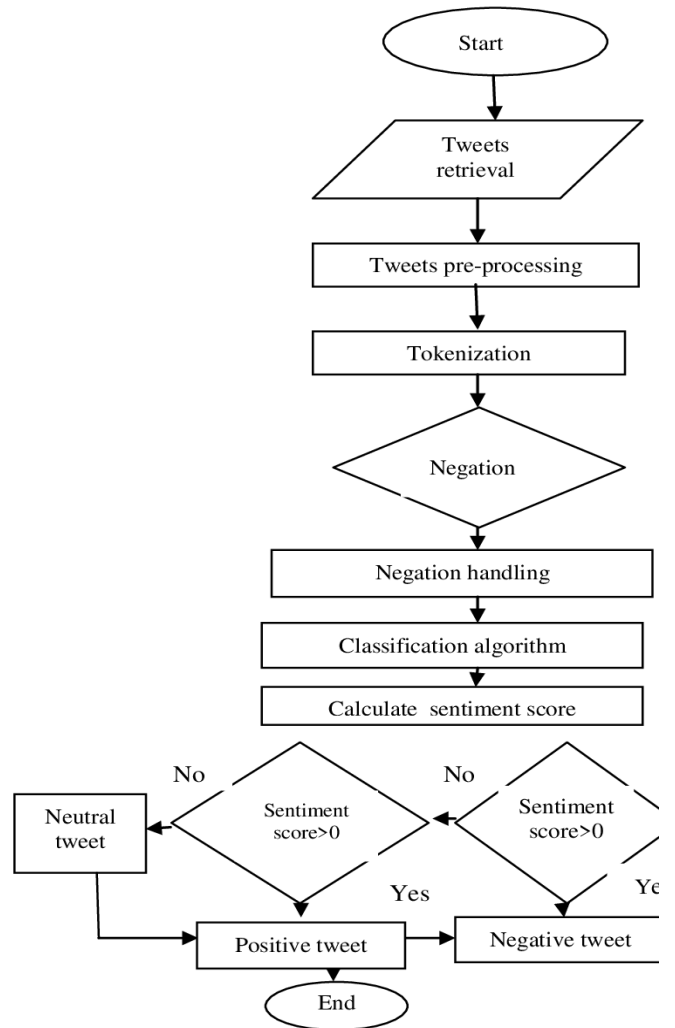


Fig 2: Data Flow Diagram

## 6.RESULT

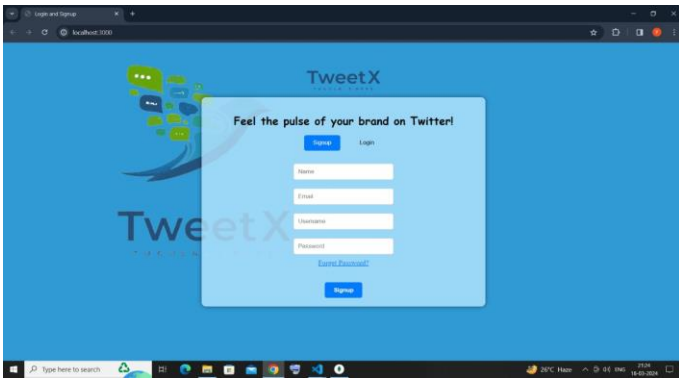


Fig 3: Signup Page



Fig 4: Login page

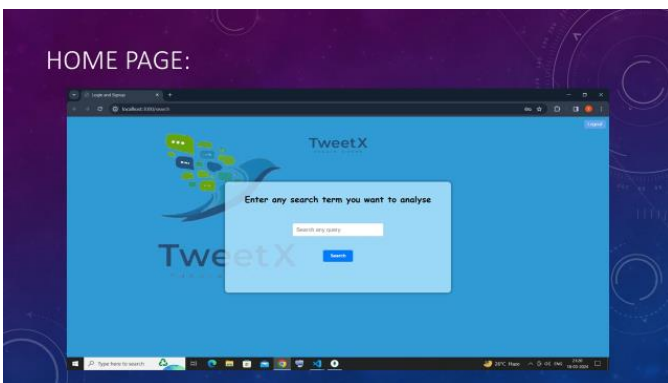


Fig 5: Home page



Fig 6: Result

## 7. CONCLUSION

In conclusion, Twitter pole analysis is a powerful tool with a big range of applications and a promising future. It allows businesses, governments, researchers, and individuals to gain valuable insights into public opinion and emotions expressed on the platform. By harnessing the vast amount of data available on Twitter, sentiment analysis provides numerous benefits.

## REFERENCE

- [1] M. Arlovski and L. Bottou. Towards principled methods for training generative adversarial networks. In ICLR, 2017.
- [2] A. Brock, T. Lim, J. M. Ritchie, and N. Weston. Neural photo editing with introspective adversarial networks. In ICLR, 2017.
- [3] T. Che, Y. Li, A. P. Jacob, Y. Bagnio, and W. Li. Mode regularized generative adversarial networks. In ICLR, 2017.
- [4] X. Chen, Y. Duan, R. Houthoofd, J. Schulman, I. Sutskever, and P. Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In NIPS, 2016.
- [5] E. L. Denton, S. Chintala, A. Szlam, and R. Fergus. Deep generative image models using a laplacian pyramid of adversarial networks. In NIPS, 2015.
- [6] C. Doersch. Tutorial on variational autoencoders. arXiv:1606.05908, 2016.
- [7] J. Gauthier. Conditional generative adversarial networks for convolutional face generation. Technical report, 2015.
- [8] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In CVPR, 2016.
- [9] X. Huang, Y. Li, O. Poursaeed, J. Hopcroft, and S. Belongie. Stacked generative adversarial networks. In CVPR, 2017.