International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Twitter Sentiment Analysis Using Machine Learning

Mr. M.K.D.S. Prasadu[1], Vasala Devi Srija[2], Vepadi Gowri Venkata Lakshmi[3], Panchadarla Srinu[4], Veedhuluri Akhilesh[5]

[1]Assistant Professor, Cse Department, Vsm College Of Engineering, Ramachandrapuram, Dr.B.R.Ambhedkar Konaseema Dt, Andhra Pradesh - 533255
[2,3,4,5]Student, Cse Department, Vsm College Of Engineering, Ramachandrapuram, Dr.B.R.Ambhedkar Konaseema Dt, Andhra Pradesh - **533255**

ABSTRACT

The analysis of public opinion has grown in importance as a result of the quick development of web technology and the growing amount of data produced on social networking sites. One of the most widely used social media sites, Twitter, offers a multitude of information with a wide range of viewpoints that are frequently unstructured, heterogeneous, and text-based. In order to categorize tweets into good, negative, or neutral sentiments according to their polarity, this research focuses on sentiment analysis on Twitter.

Web technologies and machine learning methods are used to do this. To accurately forecast attitudes, supervised classification techniques such as Support Vector Machine (SVM), K-Nearest Neighbor (KNN), Naïve Bayes, Logistic Regression, Decision Tree, and Random Forest are used. Because it can handle high-dimensional data and produce precise predictions, SVM has been recognized as one of these algorithms' most reliable features. developed with the Django framework and Python, guaranteeing a dynamic and scalable online interface for sentiment analysis. For comparison, other frameworks like as Pyramid and Flask are investigated. The created system shows how large volumes of Twitter data may be processed using machine learning algorithms and contemporary web technologies, offering insightful information to both public and private parties. Additionally, this study determines the best sentiment prediction algorithm, facilitating improved decision-making based on social media trends.

INTRODUCTION

A Twitter Sentiment Analysis project is using the Twitter API to retrieve tweets on a certain subject, occasion, or entity from Twitter and examining the sentiment conveyed by these tweets. Depending on the emotions expressed in the text, the tweets are to be categorized as either positive, negative, or neutral. The project usually entails a number of stages to do this: gathering tweets using Tweepy and other similar programs, cleaning and preparing the text (removing URLs, stop words, and special characters), and then utilizing sentiment analysis techniques. Lexicon-based methods (like VADER) or machine learning models (like SVM, Naive Bayes, or deep learning models like BERT) can be used for sentiment analysis.The model predicts the emotion of fresh tweets once it has been trained, and the findings are frequently displayed using word clouds or bar charts to offer insights. This kind of project can be used for

a number of purposes, including monitoring public opinion toward a brand, product, political figure, or event. It can even be implemented in real-time to continuously monitor Twitter conversations.

EXISTING SYSTEM

Current Twitter sentiment analysis systems usually use the Twitter API to gather historical or real-time tweets based on user accounts, hashtags, or particular keywords. After that, these systems preprocess the tweets by tokenizing them and eliminating stop words, special characters, and URLs. To categorize tweets as positive, negative, or neutral, sentiment analysis is done using either lexicon-based techniques (like VADER), machine learning models (like Naive Bayes, SVM), or deep learning models (like BERT). Tools like word clouds, bar charts, or time series graphs are frequently used to illustrate the sentiment results in order to monitor trends in sentiment. A lot of current systems also provide real-time monitoring and dashboard integration for ongoing sentiment analysis, which may be used for event analysis,
public opinion tracking, and brand monitoring. These solutions are made to be scalable, manage high tweet quantities, and give corporations and organizations useful information.

PROBLEM STATEMENT

Businesses, governments, and scholars are finding it more and more useful to analyze public sentiment toward events, companies, or personalities as a result of the exponential increase of user-generated material on social media platforms like Twitter. However, it can be difficult to glean valuable insights from tweets because of the informal language, slang, emoticons, and acronyms used.Creating a machine learning-based sentiment analysis system that can categorize tweet sentiment into positive, negative, and neutral categories is the goal of this project. In order to train a prediction model, the system will gather and preprocess tweets, translate the textual data into numerical characteristics, and use supervised machine learning methods.The results will support decision-making processes in real-time scenarios, enhance brand monitoring, improve customer feedback analysis, and better understand public sentiment.

PROPOSED SYSTEM

The suggested method uses machine learning techniques to analyze tweet sentiment. It starts by leveraging APIs to gather tweets from Twitter based on particular keywords or topic-related hashtags. Following that, these tweets undergo preprocessing to eliminate extraneous elements including URLs, mentions, hashtags, emojis, and special characters. Tokenization and lemmatization are used, stop words are eliminated, and the text is converted to lowercase as an additional cleaning step. Following preprocessing, methods such as Word Embeddings and TF-IDF are used to transform the cleaned text into numerical representations. Machine learning models like Naive Bayes, Support Vector Machine, or Logistic Regression are then trained using the converted data. The models are trained on a labeled dataset that contains neutral, negative, and positive attitudes. To choose the top-performing model, the models' performance is assessed using common measures such as accuracy, precision, recall, and F1-score. After training, the model can identify the sentiment of fresh, unseen tweets from a static dataset or in real-time. To help viewers grasp the overall sentiment distribution, the data are displayed as word clouds, pie charts, or bar graphs. Applications such as tracking political opinions, product reviews, and brand monitoring can be added to this system.

IMPLEMENTED ALGORITHMS

Support Vector Machine(SVM)

A supervised learning technique for classification and regression problems is called Support Vector Machine. It operates by determining which hyperplane in a high-dimensional space optimally divides data points of various kinds. SVM is resilient and accurate because it concentrates on the support vectors—the data points that are closest to the decision border. It uses kernel functions like polynomial or radial basis function (RBF) to handle both linear and non-linear data. SVM does well in tasks involving picture recognition and text categorization. But for big datasets, it can be computationally costly. Despite this, its great accuracy and capacity to handle complicated data continue to make it a popular option.

K-Nearest Neighbour(KNN)

A straightforward instance-based learning technique for regression and classification is K-Nearest Neighbor. It stores all training data and uses the majority label of its 'k' nearest data points to forecast the class of a new instance. Closeness is measured using distance metrics such as the Manhattan or Euclidean distance. KNN uses a lot of memory because it is simple to build and doesn't require a training phase. It can be sluggish with huge datasets but does well with smaller ones. In order to prevent underfitting or overfitting, it is crucial to select a suitable value for "k." Pattern recognition and recommendation algorithms make extensive use of it.

Naïve Bayes

Based on Bayes' Theorem, Naïve Bayes is a probabilistic machine learning algorithm. Given the class label, it makes the "naïve" assumption that features are conditionally independent of one another, which turns out to be quite effective in reality. Given the input features, it determines the likelihood of each class and chooses the one with the highest probability. For text classification applications like sentiment analysis and spam detection, Naïve Bayes works very well. It functions effectively with high-dimensional data and is quick and easy to use. It works effectively with big datasets and produces trustworthy results in spite of its strong presumptions.

Logistic Regression

A statistical technique for binary and multiclass classification issues is logistic regression. In contrast to linear regression, it maps predicted values to probabilities between 0 and 1 using a logistic (sigmoid) function. It simulates how input features and the likelihood of a specific result are related. When there is a linear relationship between features and output, logistic regression is simple to use, interpretable, and effective. Applications such as fraud detection, disease prediction, and customer churn analysis frequently employ it. It is a good foundation model, however it might not work well with complex or non-linear data.

Random Forest

A decision tree is a model that resembles a flowchart and is used for both regression and classification problems. Using conditions that result in a decision, it divides the data into branches according to feature values. A test on a feature is represented by each internal node, the test's result by each branch, and a class label or output value by each leaf node. Decision trees don't require feature scaling and are simple to comprehend and analyze. They can, however, overfit, particularly when dealing with noisy data. Pruning is one technique that can help them function better. Applications such as medical diagnosis and loan approval systems benefit from their utilization.

Decision Tree

Several decision trees are constructed using the Random Forest ensemble learning technique, which then aggregates the results to produce a final forecast. It is more precise and stable than a single decision tree

because it minimizes overfitting by averaging the output of multiple trees. To increase variety and decrease bias, each tree in the forest is trained using a random selection of characteristics and data. Both classification and regression tasks can be successfully handled by Random Forest. It can manage outliers and missing values and works well with big datasets. Predictive analysis is one of its many applications in marketing, banking, and healthcare.

WORKING OF MACHINE LEARNING ALGORITHMS

The Support Vector Machine (SVM) handles non-linear data by using kernel methods to determine the best hyperplane that divides data points of various classes with the largest margin. K-Nearest Neighbor (KNN) uses distance measurements like Euclidean distance to store the full training dataset and classify new data according to the majority class of its "k" closest neighbors. Assuming all attributes are independent, Naïve Bayes determines the likelihood that a data point belongs to a class using Bayes' Theorem and chooses the class with the highest probability. A sigmoid function, which converts predictions to values between 0 and 1, is used in logistic regression to describe the connection between input data and the likelihood of a target class. Using criteria, the choice Tree divides the dataset into subgroups according to feature values, creating a tree with each leaf signifying a final choice. To increase accuracy and decrease overfitting, Random Forest constructs several decision trees using arbitrary subsets of data and features, then aggregates their predictions. KNN is non-parametric and depends on similarity, whereas SVM and Logistic Regression work well with data that can be divided linearly. While Random Forests and Decision Trees are excellent for managing intricate interactions, Naïve Bayes works best for probabilistic models with straightforward assumptions. Every algorithm has advantages and disadvantages and is selected according to the problem and the type of data.

SYSTEM ARCHITECTURE

The architecture of the Twitter Sentiment Analysis system is modular. It starts with a Data Collection module that collects tweets over the Twitter API. The data is cleaned by the Preprocessing module, which also normalizes the text and eliminates unnecessary stuff. The cleaned text is transformed into numerical vectors in the Feature Extraction module by employing techniques such as TF-IDF or word embeddings (e.g., Word2Vec, BERT). These vectors are submitted to the Machine Learning module, which classifies sentiment using models such as SVM, LSTM, or Logistic Regression. Metrics such as F1-score and accuracy are used to assess the model. The model is used for real-time sentiment classification in a Real-time Inference Pipeline after it has been trained. Dashboards are used by a visualization layer to display the findings. The backend services that oversee the system include FLASK

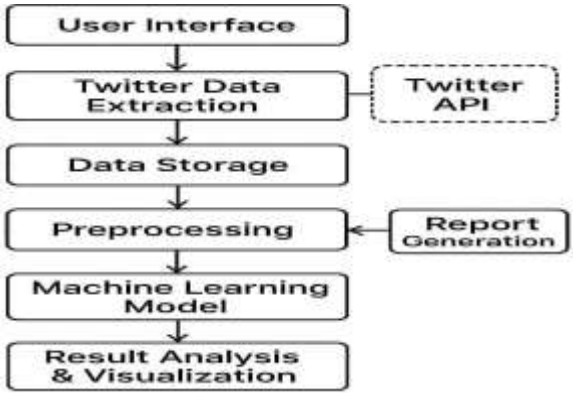

Fig:1 System Architecture Diagram

1. User Interface:

The front-end platform that users use to interact with the system is called the user interface (UI). It enables users to choose filters, enter search parameters, and start the Twitter data extraction procedure. Creating a user-friendly experience and guaranteeing seamless communication between the user and the backend system depend on this stage.

2. Twitter Data Extraction:

3. The Twitter API, which gives access to tweets based on keywords, hashtags, user accounts, or timeframes, is used in this stage to gather Twitter data. The system can retrieve vast amounts of historical or real-time tweets thanks to this API. It serves as the entry point for obtaining unprocessed social media data for analysis

3.Data Storage:

Following extraction, the data is organized for convenient access and subsequent processing. Databases or cloud storage options may be used in this situation. Effective data organization guarantees improved performance later on, particularly when working with big datasets.

4. Preprocessing:

One important stage is preprocessing, which involves cleaning and converting the raw data into a format that can be used. This entails deleting duplicates, resolving missing values, transforming text into structured formats, and removing extraneous stuff. The machine learning models' efficiency and accuracy are increased in this step.

5. Machine Learning Model:

This step involves feeding the preprocessed and cleaned data into a machine learning model. The model may carry out sentiment analysis, subject classification, trend prediction, or other NLP tasks, depending on the use case. The model generates valuable insights by learning from the data patterns.

6. Result Analysis & Visualization:

Charts, graphs, and dashboards are used to examine and visualize the model's output. This step aids users in comprehending the data's main conclusions, trends, and patterns. For simpler interpretation, a Report Generation component can optionally automatically provide thorough summaries based on the results.

RESULT ANALYSIS

Fig:2 Data Set

Fig:3 Tweet Data

Fig:4 Admin login

Fig:5 Admin login

Fig:6 Classification report

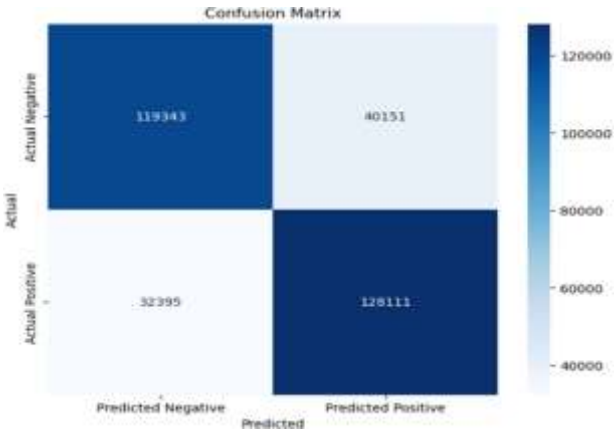

Fig:7 Confusion Matrix

Fig:8 User Login

Fig:9 Positive Tweet

Fig:10 Negative Tweet

CONCLUSION

Based on real-time social media data, the Twitter Sentiment Analysis project offers a potent foundation for comprehending trends and public opinion. It effectively harvests massive amounts of tweets, cleans them using data cleaning methods, and uses machine learning models to assess attitudes by utilizing the Twitter API. By combining report generation with visualization tools, interpretability is improved and consumers are empowered to make data-driven choices. Businesses, researchers, and policymakers can all benefit greatly from this system's ability to analyze opinion shifts, monitor public reaction, and respond appropriately. All things considered, the study demonstrates how AI and data science can be used practically to extract valuable insights from unstructured social media content.

REFERENCES

1. Twitter Developer Documentation Twitter API. (n.d.). *Twitter Developer Platform*. Retrieved from: https://developer.twitter.com/en/docs
2. Natural Language Toolkit (NLTK) Bird, S., Klein, E., & Loper, E. (2009). *Natural Language Processing with Python*. O'Reilly Media. https://www.nltk.org/
3. Scikit-learn: Machine Learning in Python Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011). *Scikit-learn: Machine Learning in Python*. Journal

of Machine Learning Research, 12, 2825–2830. https://scikit-learn.org/

4. Pandas: Python Data Analysis Library McKinney, W. (2010). *Data Structures for Statistical Computing in Python*. Proceedings of the 9th Python in Science Conference, 51-56. https://pandas.pydata.org/
5. Matplotlib and Seaborn for Data Visualization Hunter, J. D. (2007). *Matplotlib: A 2D Graphics Environment*. Computing in Science & Engineering, 9(3), 90–95. Waskom, M. et al. (2021). *Seaborn: Statistical data visualization*. https://matplotlib.org/ https://seaborn.pydata.org/
6. TextBlob: Simple Text Processing Loria, S. (2018). *TextBlob: Simplified Text Processing*. https://textblob.readthedocs.io/en/dev/
7. Kaggle Datasets (Optional if used for sample data) *Kaggle: Datasets for Twitter Sentiment Analysis*. Retrieved from: https://www.kaggle.com/