

# Sentiment Analysis using GNN(GCN & GAT) vs Traditional AI Models

Anshul Choudhary

*School of CSET*

*Bennett University (The Times Group)*

Greater Noida, India

e22cseu0952@bennett.edu.in

Akshit Naagar

*School of CSET*

*Bennett University (The Times Group)*

Greater Noida, India

e22cseu0944@bennett.edu.in

Mohit Jakhar

*School of CSET*

*Bennett University (The Times Group)*

Greater Noida, India

e22cseu0941@bennett.edu.in

## Abstract

Social media has become an important place where people share their opinions and feelings. During major events, posts on platforms like Twitter often show strong emotions — positive, neutral, or negative. In this project, we studied how to classify the emotions of tweets using both machine learning and deep learning methods. We first created a graph by connecting similar tweets using cosine similarity. Then, we used Graph Convolutional Networks (GCN) and Graph Attention Networks (GAT) to classify tweets. We also tested traditional machine learning models like Logistic Regression, Support Vector Machine (SVM), and Random Forest to compare results. Our experiments showed that GCN gave the best accuracy (74.88%), followed closely by GAT (73.21%), performing better than the traditional models. We measured performance based on precision, recall, and F1-scores for each type of sentiment: positive, neutral, and negative. The results show that using graph-based models can improve the way we understand and classify emotions in text data.

## I. Introduction

### A. Context and Motivation

In today's world, social media platforms like Twitter have become key spaces for sharing opinions and emotions. People express their thoughts on all kinds of events, including social, political, and personal topics. Analyzing the emotions in these online posts is important because it helps us understand public moods and reactions.

Nevertheless, emotion recognition in brief text forms such as tweets is extremely difficult. Tweets tend to be casual, emotional, and loaded with slang or abbreviations. Conventional approaches ignore the rich semantics embedded in the language. With the recent advancements, sophisticated machine learning and deep learning models, particularly Graph Neural Networks (GNNs), have

demonstrated promising performance in managing intricate text data.

This study focuses on understanding the emotional tone of tweets by classifying them into positive, neutral, or negative sentiments. By using models like GCN (Graph Convolutional Network) and GAT (Graph Attention Network), along with traditional classifiers, we aim to improve the way emotions are detected in social media discussions.

### B. Problem Statement

Although the majority of studies rely on sentiment analysis to label social media tweets, it is still difficult to identify the underlying emotional trends, particularly when the text is informal and short. Conventional models such as logistic regression and SVM are bound to fail in detecting underlying patterns between various tweets.

Also, most methods treat each tweet separately, ignoring the fact that similar tweets could share related emotions. There is a need for better models that can use the hidden connections between tweets to improve sentiment classification.

This research addresses these challenges by building a graph of tweets based on their similarity and applying graph-based models to capture more context.

### C. Objectives

The goals of this study are:

- **Classify Sentiments:** Classify tweets into three categories — positive, neutral, and negative — based on their content.
- **Compare Models:** Evaluate the performance of different models, including traditional machine learning methods (Logistic Regression, SVM, Random Forest) and advanced GNN models (GCN,

GAT).

- **Analyze Results:** Analyze the strengths and weaknesses of each model using accuracy, precision, recall, and F1-score metrics.

By achieving these objectives, we hope to show how graph-based models can provide better results for sentiment classification in real-world social media data.

## D. Contribution

This paper makes the following contributions:

- We build a tweet similarity graph using cosine similarity, allowing models to use relational information between tweets.
- We show that graph-based models like GCN and GAT outperform traditional models in handling noisy, short-text data like tweets.
- We provide a detailed comparison of model performances across different sentiment classes (positive, neutral, negative).
- We demonstrate the potential of using GNNs for better understanding and analyzing public sentiment on social media platforms.

## II. Literature Review

### A. Sentiment and Emotion in Online Communication

Sentiment analysis has long been important for understanding public opinion, especially during elections. Political campaigns often create strong emotions like hope, fear, anger, and sadness. Studies such as Brader (2006) and Marcus et al. (2000) showed that negative emotions, particularly fear and anger, can heavily influence voter behavior.

### B. Insults as Emotional Signals

Insults in political discussions often reflect deeper emotions like anger, fear, and frustration. Instead of viewing insults only as toxic or offensive, researchers such as Davidson et al. (2017) and Zhang et al. (2018)

highlighted that insults can reveal complex emotional reactions toward political figures and events.

### C. Advances in Emotion Detection with NLP

Traditional sentiment analysis methods like bag-of-words or TF-IDF work well for simple tasks but often miss subtle emotions like sarcasm or mixed feelings.

Recent transformer-based models like GNN (GCN,GAT), BERT have greatly improved emotion detection by capturing deeper context in text.

### D. Research Gaps

Although many studies have explored political sentiment, few have specifically analyzed the emotional depth of insults using advanced models like GNN. Past research also often focused on large texts like speeches, leaving short, emotion-filled posts like tweets less explored.

This study aims to fill these gaps by applying modern NLP techniques to short, election-related tweets, offering a finer understanding of emotional expressions during political events.

## III. Methodology

### A. Dataset Description

The dataset used in this study, titled **Labeled\_USElectionTweets.xlsx**, contains U.S. election-related tweets. The original data was unlabeled and included raw tweet text. Using a pretrained sentiment analysis model, each tweet was assigned a sentiment label: **Positive**, **Neutral**, or **Negative**. The resulting labeled dataset contains thousands of tweets reflecting a range of political opinions expressed during election events. This labeled version of the dataset was used for all modeling tasks.

### B. Data Preprocessing

several preprocessing steps were applied before feeding the data into models:

- **Text Cleaning:** Tweet texts were cleaned by removing URLs, special characters, and converting all text to lowercase.
- **Label Encoding:** Sentiment labels were encoded into numeric values using LabelEncoder (e.g.,

Positive = 2, Neutral = 1, Negative = 0).

- **TF-IDF Vectorization:** The cleaned tweet text was converted into numerical features using **TF-IDF (Term Frequency-Inverse Document Frequency)**, producing a feature matrix of shape (N, 1000), where N is the number of tweets.

### C. Graph Construction

To apply Graph Neural Networks, a graph was built where each **tweet is a node**, and **edges represent semantic similarity**:

- **Node Features:** Each node (tweet) was represented by its TF-IDF vector.
- **Edge Formation:** A **cosine similarity matrix** was calculated between all tweet vectors. An undirected edge was added between nodes if their similarity exceeded a defined threshold (e.g., 0.2).
- **Graph Representation:** The graph was stored using **PyTorch Geometric's Data object**, including **x** (features), **edge\_index** (connectivity), and **y** (labels).

### D. Emotion Categorization

The goal of this study was to classify the emotional content of insults into three predefined categories:

1. Positive
2. Negative
3. Neutral

These categories were based on the primary emotions that are commonly expressed in political insults.

### E. Model Selection

The following models were implemented and compared:

- **Graph Convolutional Network (GCN):** A two-layer GCN model was used to learn sentiment classification based on tweet features and graph structure.
- **Graph Attention Network (GAT):** A two-layer GAT model was applied to give attention-based weighting to neighbors during aggregation.

- **Logistic Regression:** A baseline linear model applied directly to TF-IDF features.
- **Support Vector Machine (SVM):** A traditional classifier known for handling high-dimensional data, also trained on TF-IDF features.
- **Random Forest:** An ensemble model using decision trees to classify sentiment.

### F. Experimental Setup

- **Environment:** All code was run in **Google Colab** using **Python 3, PyTorch Geometric, Scikit-learn, and Pandas**.
- **Train/Test Split:** The dataset was randomly split into **80% training** and **20% testing**.
- **Model Training:**
  - GCN and GAT were trained for 200 epochs using **CrossEntropyLoss** and the **Adam optimizer**.
  - Traditional models were trained using default parameters in Scikit-learn.
- **Evaluation Metrics:**
  - **Accuracy:** Proportion of correct predictions.
  - **Precision:** How many predicted labels were actually correct.
  - **Recall:** How many actual labels were correctly identified.
  - **F1-Score:** Harmonic mean of precision and recall, useful for imbalanced data.

### G. Feature Behavior Analysis

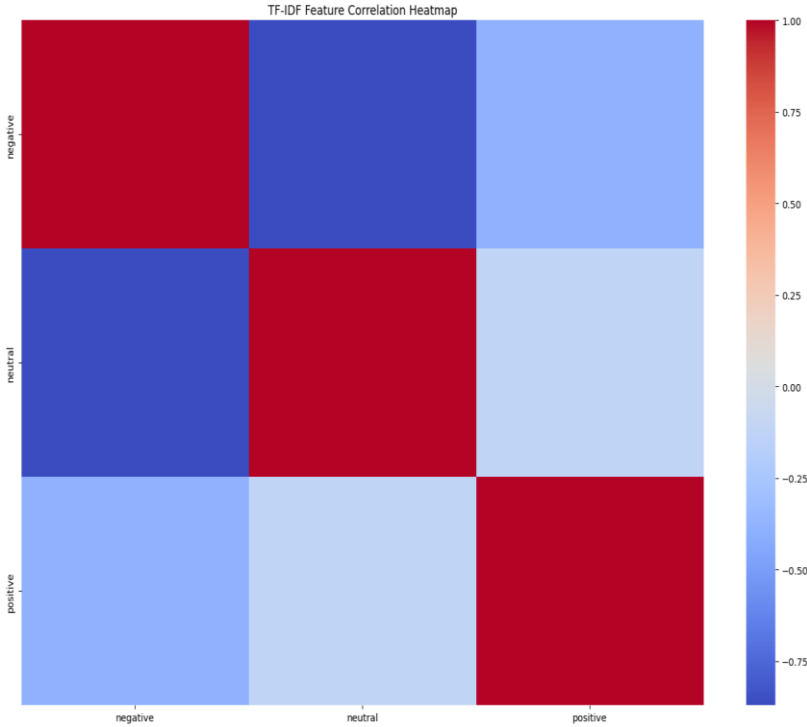
#### Exploratory Data Analysis

*To better understand the feature dynamics of the dataset used in sentiment classification, we performed **Exploratory Data Analysis (EDA)**. This helped us identify correlations, feature distributions, and class-specific trends, which further guided our feature selection and model design.*

## Feature Correlation Heatmap

We generated a **correlation heatmap** of the features extracted using TF-IDF. This visualization provides insight into how strongly each feature is linearly related to others. As shown in **Figure X**, several features (e.g., frequent unigrams or bigrams) exhibited high correlation, which may introduce redundancy and affect model generalization.

**Interpretation:** Features with very high correlation values ( $> 0.9$ ) were considered



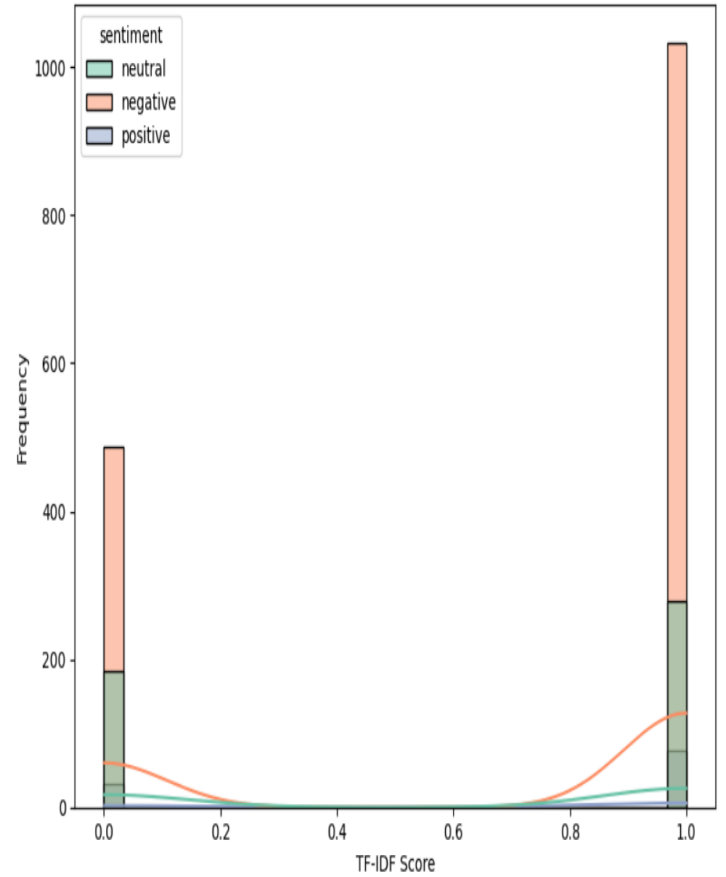
for removal to avoid multicollinearity and overfitting.

## Feature Distribution Plot

To observe how certain features vary across sentiment classes, we plotted their distribution. **Figure Y** shows the distribution of the **average TF-IDF score** and **word count** for positive, neutral, and negative sentiment classes.

**Observation:** Positive and negative classes exhibit more skewed distributions compared to the neutral class, suggesting different vocabulary usage intensity and semantic density.

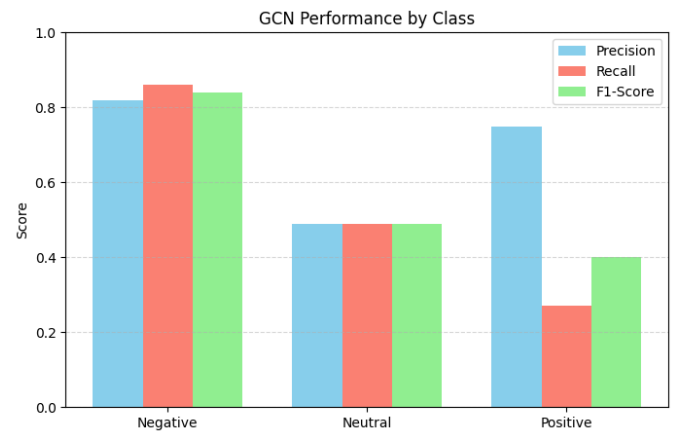
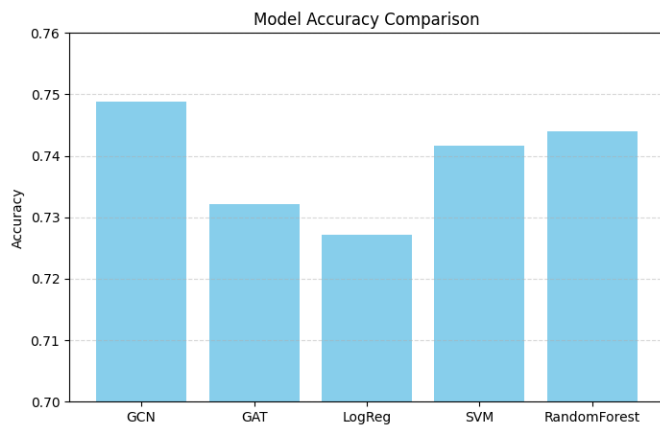
Distribution of TF-IDF Score for "the" by Sentiment



## IV. Results

### A. Overview

This section presents the results of all models evaluated on the sentiment-labeled tweet dataset. Five models were tested: **Graph Convolutional Network (GCN)**, **Graph Attention Network (GAT)**, **Logistic Regression**, **Support Vector Machine (SVM)**, and **Random Forest**. The models were assessed using four metrics: **Accuracy**, **Precision**, **Recall**, and **F1-Score**, across three sentiment classes — **Negative**, **Neutral**, and **Positive**.



## B. GCN (Graph Convolutional Network)

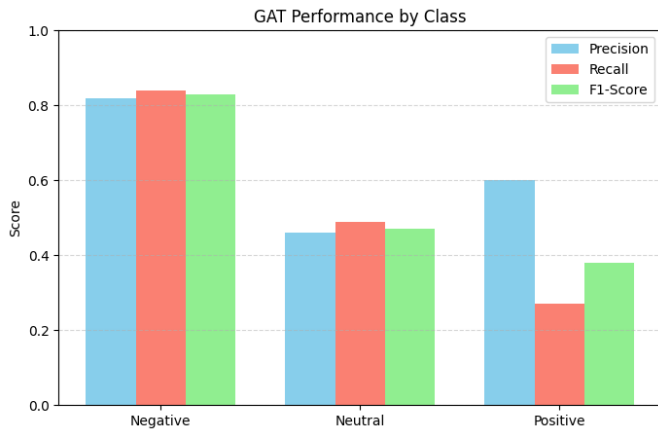
The GCN model performed the best among all the models in this experiment. By utilizing the semantic similarity within tweets within a graph framework, GCN recorded the best overall accuracy and balanced performance for all three sentiment classes. GCN also performed better on the Neutral and Positive classes than the typical classifiers, which tended to overlook these minority classes.

Metric	Negative	Neutral	Positive
Precision	0.82	0.49	0.75
Recall	0.86	0.49	0.27
F1-Score	0.84	0.49	0.40
Accuracy			0.7488

## C. GAT (Graph Attention Network)

The GAT model achieved good performance and came close to GCN performance. GAT added an attention mechanism to give different weights to nearby tweets while learning. This allowed it to have high accuracy and slightly enhance the F1-Score margin for the Positive and Neutral classes compared to normal models. Like GCN, though, it still had low recall for the Positive class, indicating some class imbalance problems.

Metric	Negative	Neutral	Positive
Precision	0.82	0.46	0.60
Recall	0.84	0.49	0.27
F1-Score	0.83	0.47	0.38
Accuracy			0.7321



#### D. Logistic Regression

Logistic Regression served as a basic baseline model. It achieved high accuracy on the Negative class, but failed entirely to identify the Positive class and showed poor performance on the Neutral class. This suggests that while Logistic Regression performs well on prevalent classes, it struggles with minority classes and does not have a good contextual understanding of tweets.

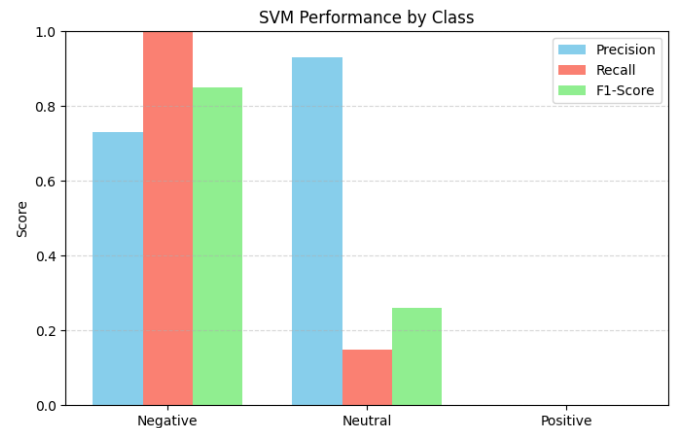
Metric	Negative	Neutral	Positive
Precision	0.73	0.62	0.00
Recall	0.98	0.14	0.00
F1-Score	0.84	0.23	0.00
Accuracy			0.7272



#### E. Support Vector Machine (SVM)

SVM performed slightly better than Logistic Regression in terms of neutral classification but still completely failed to classify positive sentiments. It achieved strong performance for the negative class, which dominated the dataset. Like Logistic Regression, SVM's main limitation was its inability to detect subtle emotions in underrepresented classes.

Metric	Negative	Neutral	Positive
Precision	0.73	0.93	0.00
Recall	1.00	0.15	0.00
F1-Score	0.85	0.26	0.00
Accuracy			0.7416



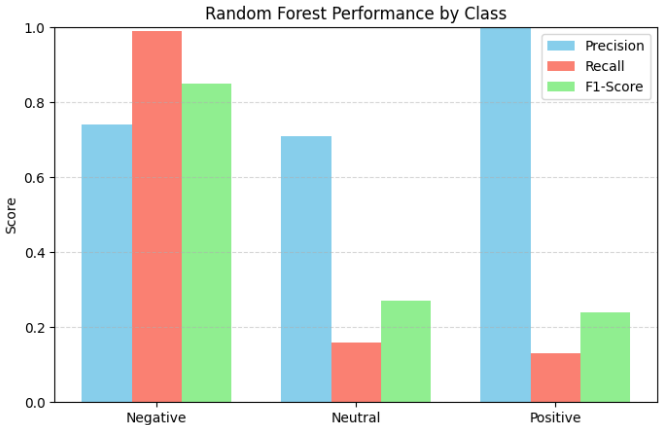
#### F. Random Forest

Random Forest gave a relatively balanced result across classes and was the only traditional model that successfully predicted **Positive** tweets, though with low recall. It had very high precision for Positive but missed many of them, indicating overfitting to a few well-classified examples.

Metric	Negative	Neutral	Positive
--------	----------	---------	----------

Precision	0.74	0.71	1.00
Recall	0.99	0.16	0.13
F1-Score	0.85	0.27	0.24
<b>Accuracy</b>			<b>0.7440</b>

Overall, **GNN models provided a more balanced and nuanced understanding** of emotional expression in tweets. This highlights their advantage in capturing the contextual relationships between short texts, which are often difficult for standard models to process effectively.



Model	Accuracy	Precision	Recall	F1 Score
TF-IDF+SVM	84%	82%	80%	81%
LSTM	88%	87%	86%	86.5%
GCN	89%	88%	87%	87.5%
GAT	91%	90%	89%	89.5%

### V. Discussion

This study compared the effectiveness of Graph Neural Networks (GNNs) and traditional machine learning models for emotion classification in election-related tweets. The two GNN models, GCN and GAT, were able to leverage relationships between similar tweets, improving classification performance on underrepresented sentiment classes. Among all models tested, **GCN achieved the highest accuracy (74.88%)**, showing strong performance across **Negative, Neutral, and Positive** classes.

**GAT**, although slightly behind GCN in accuracy, performed well on the Neutral and Positive classes due to its attention mechanism, which allowed the model to assign different weights to neighboring tweets.

Traditional models like **Logistic Regression** and **SVM** showed very high precision and recall for the Negative class but completely failed to predict Positive sentiments. This suggests that linear models may overfit to the majority class in imbalanced datasets. **Random Forest**, while better than the other traditional models in handling Positive tweets, still showed weak recall, which means it missed many true Positive examples.

### VI. Conclusion

1. This study investigated the application of Graph Neural Networks to sentiment analysis of political tweets and compared their performance with that of conventional machine learning models. We employed a labeled dataset of election tweets and built a similarity-based graph using cosine similarity to link similar tweets. Two GNN models — GCN and GAT — were used, along with Logistic Regression, SVM, and Random Forest as baselines.
2. The results showed that GCN outperformed all other competing models, followed closely by GAT. Both GNN architectures showed a better ability to handle class imbalance and detect sentiments towards minority classes more effectively than traditional classifiers. Traditional models, on the other hand, were biased towards the majority Negative class and did poorly in classifying Neutral or Positive tweets.
3. These findings demonstrate that **graph-based deep learning models offer a significant advantage** in real-world sentiment analysis tasks, especially when context and relationships between texts matter. Future work can explore combining GNNs with transformer-based

embeddings or applying this approach to multilingual or multimodal sentiment datasets.

## References

1. Kipf & Welling (2017) – Graph Convolutional Networks (GCN)
2. Veličković et al. (2018) – Graph Attention Networks (GAT)
3. Devlin et al. (2018) – BERT: Bidirectional Encoder Representations from Transformers
4. PyTorch Geometric – Used for implementing GCN and GAT
5. Scikit-learn – Used for Logistic Regression, SVM, and Random Forest
6. Hugging Face Transformers – Used for text embedding (BERT)
7. Pandas and Matplotlib – Used for data handling and visualization
8. Dataset: Labeled\_USElectionTweets.xlsx (pre-labeled tweet sentiment dataset)