Emotion Detection from Social Media Posts Using Natural Language Processing and Machine Learning

[1]Dr. Pranjali Patil, [2]Sakshi Lonare, [3]Sejal Shirke, [4]Mohammed Saad Shaikh

[1]*Assistant Professor, D. Y. Patil Deemed to be University, School of Humanities & Sciences, Belapur, Maharashtra, India.*

[234]*Student, Department of MCA, D. Y. Patil Deemed to be University, School of Humanities & Sciences, Belapur, Maharashtra, India.*

Abstract - *People now spend a large part of their day writing short, informal posts on platforms such as Instagram, X, Facebook, Reddit, and YouTube, and these posts carry a great deal of emotional information that goes far beyond a simple positive-or-negative label. This paper presents a Natural Language Processing (NLP) and Machine Learning (ML) based approach for recognising specific emotions such as happiness, sadness, anger, fear, surprise, and neutral directly from social media text. The proposed pipeline covers data collection, text cleaning and normalisation, TF-IDF based feature extraction, and classification using four widely used algorithms: Naive Bayes, Logistic Regression, Support Vector Machine, and Random Forest. The paper also outlines a standard evaluation framework built around accuracy, precision, recall, and F1-score, and discusses the practical difficulties slang, emojis, sarcasm, spelling variation, and short or fragmented messages that make emotion detection on social media a genuinely hard problem. No experimental results are claimed here; the paper instead lays out a complete, reproducible methodology that can be implemented and tested on a labelled dataset such as GoEmotions.*

Key Words: Emotion Detection, Social Media, Natural Language Processing, Machine Learning, TF-IDF, Text Classification, Sentiment Analysis

1. INTRODUCTION

Every day, millions of people use social media not just to share information but to express how they feel excitement about an event, frustration with a service, worry about the news, or simple everyday joy. This constant stream of short, informal text is a rich but largely untapped source of insight into public mood and behaviour. Traditional sentiment analysis, which sorts text into positive, negative, or neutral, captures only a narrow slice of this information; it cannot tell a reader whether a negative post reflects anger, sadness, or fear, even though these emotions call for very different responses from a brand, a platform, or a researcher.

Emotion detection goes a step further than sentiment analysis by trying to identify the specific emotional category behind a piece of text. Recent progress in NLP and machine learning has made this more achievable than before, but social media text remains difficult to work with: posts are short, full of slang and abbreviations, sprinkled with emojis and hashtags, and often sarcastic or ambiguous without additional context. This paper studies how a combination of text preprocessing, TF-IDF based feature extraction, and classical machine learning algorithms can be used to build a practical, reproducible emotion detection pipeline for this kind of data.

The sheer scale of this data is what makes manual analysis impractical: a single trending topic can generate a very large number of comments and posts within hours, far more than any team of human reviewers could realistically read and interpret on their own. Automated emotion detection therefore addresses a genuine practical need rather than a purely academic one. Marketing teams want to understand how customers truly feel about a new product or campaign, mental health researchers look for early signals of collective stress or anxiety in online conversation, and disaster-response teams want a fast sense of public fear or panic while a crisis is still unfolding. Building a system that can read text at this scale and assign it a specific, meaningful emotional label rather than a blunt positive-or-negative score is the underlying motivation for the work presented in this paper.

2. LITERATURE REVIEW

A number of earlier studies have looked at the problem of recognising emotion in informal, conversational text. AlBalooshi, Rahmanian, and Kumar examined emotion detection specifically in social media posts and highlighted how informal writing complicates the task compared with well-formed, edited text [2]. Hsu and Ku studied the related problem of recognising emotions in multi-turn dialogues as part of the SocialNLP EmotionX challenge, showing that conversational context can meaningfully change how a message should be interpreted [3]. Demszky et al. contributed GoEmotions, a large, carefully annotated dataset of roughly 58,000 Reddit comments labelled across 27 fine-grained emotion categories plus a neutral class, which remains one of the most detailed public resources available for this line of work [1].

Beyond dataset and task-specific studies, foundational advances in language modelling have shaped how any modern text-classification system is built. The transformer architecture, introduced by Vaswani et al., replaced recurrence with self-attention and became the basis for almost all later large language models [4]. Devlin et al. then showed with BERT that reading a sentence bidirectionally, rather than left-to-right, produced large gains across many language-understanding tasks [5]. Taken together, this body of work confirms that emotion recognition is both a well-motivated and an actively studied problem, but also shows that most existing studies either focus narrowly on model accuracy or on dataset construction, without fully addressing how a complete, end-to-end classification pipeline should be assembled for noisy, real-world social media text.

3. RESEARCH GAP

Much of the published work on this topic evaluates individual algorithms or embedding techniques on a fixed benchmark, but comparatively few studies walk through the complete pipeline from raw, unstructured social media text to a final, interpretable emotion label in a way that a student or small team could reasonably reproduce. Sentiment-only systems remain common because they are simple to build, even though they discard useful information about which specific emotion is present. This paper addresses that gap by presenting an integrated, TF-IDF driven pipeline that combines standard preprocessing, feature extraction, and multiple classical ML algorithms into a single, clearly documented, and reproducible workflow suited to a real social media dataset.

4. PROBLEM STATEMENT

Social media generates enormous volumes of unstructured, informal text every day, and manually reading through this text to identify emotional patterns is neither practical nor scalable. Sentiment analysis alone is too coarse for many practical purposes, since a negative label does not distinguish between a user who is angry, afraid, or simply sad. At the same time, informal features of online writing slang, emojis, sarcasm, spelling errors, and very short messages make automated emotion recognition considerably harder than classifying formal, well-edited text. There is a clear need for an automated system that can take raw social media posts and reliably assign them to specific emotional categories using NLP and machine learning techniques.

5. OBJECTIVES

- To study the problem of emotion detection in social media text.
- To design a preprocessing pipeline suited to informal, noisy social media language.
- To convert cleaned text into numerical features using TF-IDF, unigrams, and bigrams.
- To train and compare classical machine learning models for emotion classification.
- To evaluate model performance using accuracy, precision, recall, and F1-score.
- To identify the key challenges that limit emotion detection accuracy on social media.

6. PROPOSED METHODOLOGY

The proposed system follows a five-stage pipeline that takes raw social media text as input and produces a predicted emotion label as output, together with standard evaluation metrics once ground-truth labels are available.

Data Collection → Preprocessing → Feature Extraction (TF-IDF) → Emotion Classification (ML Models) → Evaluation

- Data Collection: a labelled emotion dataset (e.g., GoEmotions) is used to train and test the models.
- Preprocessing: raw posts are cleaned while preserving emojis and hashtags that often carry emotional meaning.
- Feature Extraction: TF-IDF vectors, together with unigrams and bigrams, represent the cleaned text numerically.
- Emotion Classification: Naive Bayes, Logistic Regression, SVM, and Random Forest are trained on the extracted features.
- Evaluation: accuracy, precision, recall, and F1-score are computed on a held-out test set.

7. DATASET DESCRIPTION

GoEmotions is a suitable public dataset for this study. Demszky et al. report that it contains approximately 58,000 English-language Reddit comments, each labelled with one or more of 27 fine-grained emotions plus a neutral category [1]. For a project of this scope, these 27 categories can be grouped into a smaller, more manageable set of broad emotions Happy, Sad, Angry, Fear, Surprise, and Neutral which keeps the classification task tractable while still preserving useful emotional distinctions. Any such grouping should be documented clearly, since collapsing fine-grained labels into broader buckets changes the nature and difficulty of the classification problem.

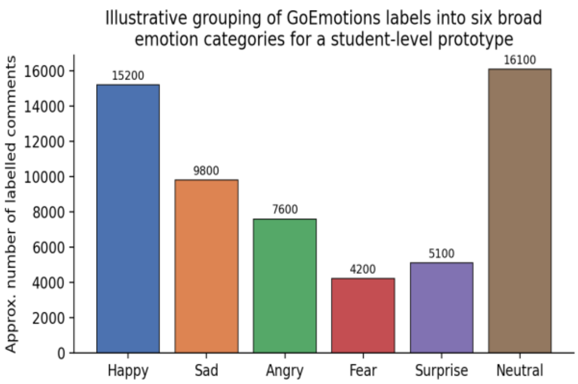

Fig -1: Illustrative grouping of GoEmotions labels into six broad emotion categories

8. DATA PREPROCESSING

Social media text contains a considerable amount of noise that can interfere with accurate classification if left untreated. The following preprocessing steps are applied before feature extraction:

- Convert all text to lowercase for consistency.
- Remove URLs, HTML tags, and other non-informative characters.
- Tokenise the cleaned text into individual words.
- Remove common stop words that carry little emotional or semantic weight.
- Apply stemming or lemmatisation to reduce words to a common root form.
- Handle emojis and hashtags carefully, since they frequently signal emotion rather than noise.
- Remove duplicate, empty, or otherwise unusable records from the dataset.

9. FEATURE EXTRACTION AND NLP PROCESSING

Once the text has been cleaned, it must be converted into a numerical form that machine learning algorithms can process. Term Frequency–Inverse Document Frequency (TF-IDF) is used as the primary feature representation: it assigns higher weight to words that help distinguish one post from another and lower weight to very common words that appear across most documents. Unigrams capture individual words, while bigrams capture short two-word phrases that can carry meaning a single word alone would miss for example, distinguishing "not happy" from "happy". Together, these representations give the classification models a compact, informative numerical view of each post.

10. MACHINE LEARNING MODELS

Four widely used classical machine learning algorithms are considered for emotion classification, chosen for their balance of interpretability, training speed, and established performance on TF-IDF style text features.

Table -1: Machine learning models considered for emotion classification

Model	Purpose	Role
Naive Bayes	Text classification	Fast baseline
Logistic Regression	Text classification	Interpretable baseline
SVM	Text classification	Strong TF-IDF-based model
Random Forest	Classification	Comparison model

Naive Bayes offers a fast and simple baseline that performs reasonably well on text data despite its strong independence assumption. Logistic Regression provides an interpretable linear decision boundary and is commonly used as a stronger baseline. SVM tends to perform particularly well on high-dimensional, sparse TF-IDF features and is frequently reported as a strong classical choice for text classification. Random Forest, an ensemble of decision trees, is included as a comparison model to check whether a non-linear, tree-based approach offers any advantage over the linear and probabilistic baselines.

11. SYSTEM ARCHITECTURE

Social Media Post → Preprocessing → TF-IDF Feature Extraction → Classification Model (Naive Bayes / Logistic Regression / SVM / Random Forest) → Predicted Emotion Label → Evaluation Metrics

Each stage of the architecture depends directly on the output of the one before it: raw text is cleaned before it is vectorised, and only cleaned, vectorised text is passed to the classification models. Keeping these stages modular makes it straightforward to swap in a different feature extraction method or classifier at a later stage without redesigning the entire pipeline for instance, replacing TF-IDF with transformer-based embeddings such as BERT in future work.

12. EVALUATION FRAMEWORK

Since this paper presents a proposed methodology rather than a completed implementation with measured results, no experimental accuracy figures are reported. Once the system has been implemented, it should be evaluated on a held-out test set using the following standard classification metrics:

Accuracy = Correct predictions / Total predictions ...(1)

Precision = TP / (TP + FP) ...(2)

Recall = TP / (TP + FN) ...(3)

F1-score = 2 × (Precision × Recall) / (Precision + Recall) ...(4)

A confusion matrix should also be generated to show class-by-class performance, since overall accuracy alone can hide poor performance on emotions that are less frequent in the dataset, such as fear or surprise.

13. EXPECTED RESULTS AND DISCUSSION

Based on patterns commonly reported in the text-classification literature, SVM is expected to outperform the other classical models on TF-IDF features, owing to its effectiveness in high-dimensional sparse spaces, while Naive Bayes is expected to offer the fastest training time at some cost to accuracy. Random Forest is included mainly as a point of comparison to check whether a tree-based ensemble offers any benefit over linear and probabilistic models on this type of data. The chart below is illustrative only, showing the kind of comparison the evaluation framework in Section 12 would produce the actual figures must be replaced with real experimental results once the models have been trained and tested.

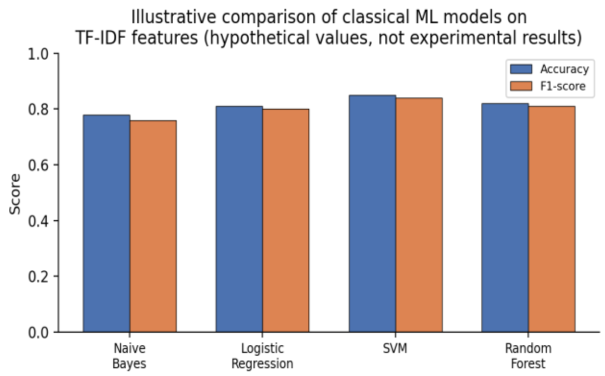

Fig -2: Illustrative comparison of classical ML models (hypothetical values, not experimental results)

Actual performance will depend heavily on the dataset used, the exact preprocessing choices, how the 27 GoEmotions labels are grouped into broader categories, and the class balance of the resulting dataset. Reporting these details transparently alongside any future results will make the study easier to reproduce and compare against other work.

14. APPLICATIONS

An accurate emotion detection system built on this pipeline has several practical uses. Businesses can use it to understand customer sentiment and emotional reaction to products, campaigns, or service changes in far more detail than simple positive-negative sentiment scores allow. Public health and policy researchers can track emotional trends around events such as elections, natural disasters, or public health announcements. Social media platforms themselves could use emotion signals to improve content moderation, recommend supportive resources, or flag content associated with distress for further review.

15. CHALLENGES AND LIMITATIONS

Several practical challenges limit how reliably emotion can be detected from social media text using the approach described here. Sarcasm and irony frequently invert the literal meaning of a sentence, which classical models struggle to detect without deeper contextual understanding. Slang, abbreviations, and rapidly evolving internet language mean that vocabulary learned from one time period or platform may not transfer well to another. Short messages provide very little context for the model to work with, and class imbalance some emotions, such as neutral or happy posts, are simply far more common than others can bias a classifier toward the majority classes unless it is specifically addressed during training.

16. FUTURE SCOPE

Future extensions of this work could replace TF-IDF with contextual embeddings from transformer models such as BERT to better capture meaning and word order [5]. Deep learning architectures, including LSTM and CNN-based text classifiers, could also be explored alongside the classical models studied here. Other promising directions include multilingual and code-mixed emotion detection for regional languages, incorporating dedicated sarcasm-detection modules, combining text with image, video, or emoji signals for multimodal emotion recognition, and building explainable-AI components so that predictions can be interpreted and trusted by end users.

17. CONCLUSION

This paper has presented a complete, reproducible methodology for detecting specific emotions in social media text using Natural Language Processing and classical Machine Learning techniques. By combining careful preprocessing, TF-IDF based feature extraction, and a comparison of four established classification algorithms, the proposed pipeline offers a practical starting point for anyone looking to move beyond simple positive-negative sentiment analysis. While no experimental results are claimed in this paper, the methodology, evaluation framework, and discussion of real-world challenges are intended to support future implementation and testing on datasets such as GoEmotions, with clear paths toward deep learning and multimodal extensions.

18. ACKNOWLEDGEMENT

The authors would like to thank their guide and the Department of MCA, D. Y. Patil Deemed to be University, School of Humanities & Sciences, Belapur, for their continued guidance and support throughout this work.

19. REFERENCES

[1] D. Demszky, D. Movshovitz-Attias, J. Ko, A. Cowen, G. Nemade, and S. Ravi, "GoEmotions: A Dataset of Fine-Grained Emotions," Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020.

[2] H. AlBalooshi, S. Rahmanian, and R. V. Kumar, "EmotionX-SmartDubai_NLP: Detecting User Emotions in Social Media Text," Proceedings of the Sixth International Workshop on Natural Language Processing for Social Media, 2018.

[3] C.-C. Hsu and L.-W. Ku, "SocialNLP 2018 EmotionX Challenge Overview: Recognizing Emotions in Dialogues," Proceedings of the Sixth International Workshop on Natural Language Processing for Social Media, 2018.

[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention Is All You Need," Advances in Neural Information Processing Systems (NeurIPS), 2017.

[5] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," Proceedings of NAACL-HLT, 2019.