Emotion Classification in Short Text Messages Using Traditional ML and CNN Models

REPORT

Presented

BY

GHOSN Joyce

BEYROUTH

July 2025

Table of Contents

1. Introduction

Emotion classification from short text is a crucial task in modern natural language processing (NLP), enabling systems to better understand human intent and emotional state. Applications range from sentiment analysis and mental health detection to customer service optimization and user behavior modeling. In a digital world saturated with user-generated content, particularly on platforms like social media and online reviews, the ability to automatically detect emotions has profound implications for both academic research and industry deployment.

The goal of this project is to build a robust, end-to-end machine learning pipeline capable of classifying emotional states from brief text inputs. The pipeline leverages both traditional machine learning and deep learning methods, working on a large, real-world dataset characterized by significant class imbalance and noisy, unstructured inputs. By developing strong preprocessing routines, selecting and optimizing relevant models, and applying systematic evaluation strategies, the project aims to provide a reliable solution to emotion detection.

The project systematically investigates and compares multiple modeling strategies, including Logistic Regression, Naive Bayes, Support Vector Machines, Decision Trees, Feedforward Neural Networks, Convolutional Neural Networks, and LSTM-based architectures. It also briefly evaluates an unsupervised approach using KMeans clustering. Throughout the process, we address important challenges such as feature engineering, hyperparameter tuning, sequence length optimization, and model evaluation on balanced datasets.

Ultimately, this project identifies the most effective model architecture for emotion classification and examines how different design decisions impact performance. The work not only informs model selection for similar NLP tasks but also contributes to the broader understanding of how machine learning systems interpret and handle emotional content in text.

2. Dataset Overview

2.1 Dataset Characteristics

The dataset analyzed in this study, "emotion_sentimen_dataset1.csv," includes roughly 839,554 text messages, each labeled with an emotion such as love, happiness, neutral, boredom, hate, and relief. The dataset is notably imbalanced, with approximately 80% of the data labeled as 'neutral,' while minority emotions such as 'boredom' have as few as 126 examples.

2.2 Dataset Justification

This particular dataset was chosen due to its substantial volume, authentic linguistic diversity, and real-world complexity typical of user-generated text. The detailed labeling of emotional states provides valuable insights for evaluating model accuracy across different emotional nuances. Additionally, the pronounced imbalance creates a realistic scenario for testing advanced data handling and balancing techniques.

3. Methodological Framework

3.1 Phase 1: Data Loading, Exploration, Cleaning, and Balancing

3.1.1 Code Implementation and Interpretation

Essential Python libraries including pandas, scikit-learn, TensorFlow, and Keras were used to create an organized analytical environment. Initial data analysis included checking for missing data, duplicate entries, and distribution of emotional labels, revealing a significant class imbalance favoring neutral labels. The text was cleaned by converting to lowercase, removing punctuation, and eliminating stopwords. Additionally, text length was analyzed to inform decisions about sequence length in later deep learning models. To address class imbalance, downsampling was applied, significantly reducing the overrepresentation of neutral messages and promoting balanced model learning.

Phase 1 successfully transformed a raw, imbalanced dataset into a clean, balanced corpus suitable for training effective and generalizable models.

3.2 Phase 2: Traditional Machine Learning Preparation

3.2.1 Code Implementation and Interpretation

Post-cleaning, the text was converted into numerical form using TF-IDF vectorization, creating a 5000-feature sparse matrix suitable for traditional machine learning algorithms. This representation effectively captured the importance of different terms for distinguishing emotional categories without excessive computational complexity.

The TF-IDF approach provided an effective, interpretable, and computationally efficient basis for training traditional classifiers.

3.3 Phase 3: Traditional Classifier Benchmarking and Hyperparameter Tuning

In addition to evaluating the core performance of each traditional machine learning model, this phase also involved systematic hyperparameter tuning using GridSearchCV. For each classifier, the most influential hyperparameters were explored to optimize macro-averaged F1-score. The tuning process was critical in striking the right balance between underfitting and overfitting, particularly in high-dimensional TF-IDF feature space. Each model's configuration was refined through cross-validation, and the best parameters were then applied to the final test set for evaluation.

This phase aimed to systematically evaluate and tune traditional machine learning classifiers on the TF-IDF representation of the balanced emotion dataset. Four models—Logistic Regression, Naive Bayes, Support Vector Machine (SVM), and Decision Tree—were selected for their widespread use in NLP tasks and varying complexity. The TF-IDF features were split into training

and test sets (80/20), and models were benchmarked using accuracy, macro-averaged precision, recall, and F1-scores.

3.3.1 Logistic Regression Interpretation

Logistic Regression served as a robust baseline, achieving a macro F1-score of 0.95. Its performance was steady across most classes, with notable strengths in well-represented labels like "love" and "happiness". However, recall dropped significantly for underrepresented classes such as "boredom" and "empty", where predictions were often biased toward more frequent classes. This behavior is expected given the linear nature of Logistic Regression and its sensitivity to class imbalance, even post downsampling. Hyperparameter tuning revealed that reducing regularization ($C = 10$) improved the model's flexibility without sacrificing generalization.

3.3.2 Naive Bayes Interpretation

The Naive Bayes classifier, though computationally efficient, underperformed relative to other models with a macro F1-score of 0.75. The core assumption of feature independence limited its capacity to model the nuanced co-occurrence of emotionally significant terms. As a result, precision and recall were especially weak for classes like "empty" and "boredom", which often depend on context and subtle phrasing. Despite its fast training time and ease of interpretation, it failed to capture the underlying semantic complexity of the dataset.

3.3.3 Support Vector Machine Interpretation

The Linear Support Vector Machine achieved the highest and most balanced performance among the traditional classifiers, reaching a macro F1-score of 0.96. It maintained high precision and recall across all emotion classes, including minority labels. The margin-based classification enabled the model to separate classes effectively in the high-dimensional TF-IDF space. Tuning the regularization parameter ($C = 1$) helped strike the right balance between bias and variance. SVM also exhibited strong robustness, showing minimal overfitting and consistent performance on both training and test sets.

3.3.4 Decision Tree Interpretation

The Decision Tree model yielded a deceptively high macro F1-score of 0.98, which was later identified as a symptom of overfitting. It achieved near-perfect scores on both frequent and rare classes, including "boredom", which only had 126 examples. This suspicious uniformity was traced back to its unrestricted growth—no pruning or depth limitations were imposed during training. As a result, the tree memorized the training data instead of learning generalizable patterns, compromising its utility on unseen data.

3.3.5 Comparative Analysis of Traditional Models

SVM and Logistic Regression emerged as the most balanced models, offering both high performance and strong generalization. SVM slightly outperformed Logistic Regression,

particularly on edge cases and rarer emotions. Naive Bayes proved too simplistic for the complexity of the task, while Decision Tree, though accurate on paper, failed to generalize. This comparison underscores the importance of interpretability and reliability, not just accuracy, when selecting models for real-world NLP tasks.

Conclusion: Phase 3 confirmed that traditional linear models can perform competitively with well-preprocessed data. It also demonstrated the value of systematic benchmarking and hyperparameter tuning. Among the traditional classifiers, SVM was the most consistent and trustworthy, whereas Decision Trees highlighted the risks of overfitting when not properly constrained.

3.4 Phase 4: Deep Learning and Unsupervised Learning Models

In this phase, deep learning architectures and one unsupervised learning method were evaluated to determine their suitability for emotion classification. Models included a Feedforward Neural Network (FFNN), Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), and KMeans clustering. The goal was to assess both supervised and unsupervised learning paradigms using the same cleaned and balanced dataset.

3.4.1 Feedforward Neural Network (FFNN) Interpretation

The FFNN architecture used a basic Embedding layer followed by GlobalAveragePooling and dense layers. It reached a validation accuracy of 95.8% and a macro F1-score of 0.90. While the model performed well on common classes like "love" and "happiness", it struggled with less frequent labels such as "boredom" and "empty", with F1-scores dropping below 0.35. This indicates a limitation in its ability to differentiate subtle emotional cues. The model trained steadily with no signs of overfitting, but its limited structure reduced its effectiveness for capturing contextual relationships in short, expressive texts. to sum up, FFNN,performed significantly worse than CNN and traditional ML models, confirming that FFNNs do not capture sequential context or local n-gram patterns in text. This highlighted the importance of choosing architectures aligned with data structure.”

3.4.2 Convolutional Neural Network (CNN) Interpretation

The CNN model delivered the strongest performance overall. It achieved a validation accuracy of 98.2% and a macro F1-score of 0.99, with individual class F1-scores consistently exceeding 0.94, including for minority classes. The convolutional filters were particularly effective at capturing local word patterns and short n-grams like "feel sad" or "very angry"—key indicators in emotion recognition. The model converged within three epochs, with validation loss stabilizing early, indicating high learning efficiency and excellent generalization. Unlike the FFNN, the CNN was able to retain positional information in sequences, which helped it outperform other models across the board.

3.4.3 Long Short-Term Memory (LSTM) Interpretation

Despite its theoretical advantage in modeling sequential data, the LSTM model underperformed. Training accuracy plateaued at 22%, with no improvement across epochs and a flat validation accuracy of 21.97%. The model failed to learn meaningful patterns from the data, likely due to a combination of short sequence lengths, overfitting risk, and class sparsity. The complexity of LSTM likely outweighed its benefits for this task, particularly when dealing with short texts lacking strong temporal dependencies.

3.4.4 KMeans Unsupervised Learning Interpretation

KMeans was tested as an unsupervised baseline using TF-IDF vectors with the number of clusters set to the number of emotion classes. Its Adjusted Rand Index (ARI) of 0.0878 was only marginally better than random assignment. This result was expected, as KMeans lacks the semantic understanding necessary to differentiate emotions from plain word frequencies. Without supervision, the model was unable to align text structure with emotional meaning, making it unsuitable for this problem space.

3.4.5 Comparative Analysis of Deep Learning Models

CNN significantly outperformed both FFNN and LSTM, especially on minority emotions. FFNN provided a simple and fast baseline but lacked depth and positional awareness. LSTM failed to converge, reinforcing that model complexity should match data characteristics. CNN's use of sliding filters allowed it to effectively capture emotion-indicating phrases, making it ideal for this application.

3.4.6 Comparative Analysis Between Traditional and Deep Learning Models

When compared to the best-performing traditional models, CNN offered a clear performance gain—both in macro F1-score and class-level reliability. While SVM achieved high accuracy, it could not match CNN's performance on edge cases and minority classes. Logistic Regression was interpretable and efficient, but CNN's ability to extract patterns in context-rich data proved superior. The neural model also generalized better without significant tuning.

Conclusion: This phase reinforced that supervised deep learning, particularly CNNs, are highly effective for emotion classification in short text. FFNN served as a good baseline, while LSTM proved overly complex and under-optimized for the task. Unsupervised methods like KMeans lacked the semantic awareness required. Overall, CNN stands out as the most practical and performant solution among all tested models.

3.5 Additional Insights and Practical Considerations

3.5.1 Evaluation Strategy and Model Design Justification

To facilitate a clear and structured comparison of model performance, the following table summarizes macro-averaged F1-scores across all major models:

Model	**Validation Accuracy**	**Macro F1-Score**	**Notes**
Logistic Regression	95.0%	0.95	Consistent, slightly weaker on rare classes
Naive Bayes	77.0%	0.75	Fast, underperforms on subtle emotions
Support Vector Machine	96.0%	0.96	Strong balance and generalization
Decision Tree	98.0%	0.98	Overfitting confirmed on minority classes
Feedforward NN (FFNN)	95.8%	0.90	Stable but weak on minority emotions
Convolutional NN (CNN)	98.2%	0.99	Best performance, strong minority coverage
LSTM	21.9%	–	Failed to learn
KMeans (Unsupervised)	–	ARI = 0.0878	No label access; performed near random

Fixed Test Set for Fair Evaluation: All models were evaluated using the same stratified 80/20 train-test split. This fixed partition was applied across traditional and deep learning models to ensure fairness in model comparison and reproducibility of evaluation metrics.

Justification for Model Architecture Choices: The input sequence length was set to 100 tokens based on the text length distribution observed during Phase 1. Over 90% of text entries fell under this threshold, making it a practical cutoff for padding and truncation. Embedding dimensions were set to 64, striking a balance between representation richness and computational efficiency for this dataset size.

3.5.2 Visual Outputs and Supporting Figures

- **Text Length Histogram:** Used to determine padding threshold for neural models.
- **Emotion Distribution Heatmap:** Confirmed rebalancing effectiveness.
- **Word Clouds:** Visualized emotion-specific vocabularies (supporting exploratory understanding).

- **Training Curves:** CNN training showed rapid convergence with early plateau in validation loss, indicating generalization. FFNN had slower improvement, and LSTM curves remained flat.

3.5.3 Reflections and Limitations

- **LSTM Failure:** The LSTM architecture did not converge, likely due to short sequence lengths and limited temporal dependencies in the data. This suggests the importance of aligning model complexity with input characteristics.
- **Overfitting in Decision Tree:** Despite high metrics, Decision Tree memorized training data, especially in minority classes. Pruning or ensembling methods could mitigate this.
- **Unsupervised Limitations:** KMeans clustering showed that frequency-based features like TF-IDF lack the semantic understanding required to group emotional expressions without supervision.
- **Embedding Quality:** Using pre-trained embeddings like GloVe or BERT might have further improved generalization, especially for rare or ambiguous expressions.
- **Bias in Labels:** Some emotions overlap (e.g., sadness vs empty), which could introduce label noise not accounted for in current metrics.

4. Conclusion

This project successfully implemented a comprehensive and technically rigorous pipeline for emotion classification from short text messages. Spanning four structured phases, the work addressed real-world challenges in natural language processing such as data imbalance, model interpretability, and performance generalization. The findings not only highlight the performance of various algorithms, but also reflect the strategic choices that contributed to their success or failure.

In **Phase 1**, we began by thoroughly exploring the dataset, identifying severe class imbalance and preparing the data through cleaning, normalization, and downsampling. These steps ensured the integrity of our training data and prevented the model from being biased toward dominant emotion classes, particularly 'neutral'. By understanding text length distributions, we also made informed decisions about padding thresholds and embedding sizes for deep learning models.

In **Phase 2**, we transitioned from raw text to structured input for traditional machine learning models by employing TF-IDF vectorization. This allowed us to convert text into sparse matrices that retained term importance, providing a clean and efficient representation for algorithms such as Logistic Regression, Naive Bayes, SVM, and Decision Trees.

Phase 3 involved benchmarking traditional classifiers and optimizing their performance via hyperparameter tuning. Logistic Regression and SVM showed strong generalization, while Decision Trees, though initially promising, demonstrated overfitting. Naive Bayes fell short due to its assumptions of feature independence. The most significant insight from this phase was that linear models like SVM can offer competitive performance, especially when backed by well-engineered features and systematic tuning.

In **Phase 4**, we advanced to neural architectures. The Convolutional Neural Network (CNN) emerged as the top-performing model, with a macro F1-score of 0.99 and high precision-recall across all classes,

including low-frequency ones like 'boredom'. FFNN served as a useful baseline, while LSTM failed to converge—highlighting the importance of choosing architectures suited to the nature of the data. Additionally, we explored KMeans clustering as an unsupervised benchmark. Its poor performance reinforced the necessity of supervision in nuanced classification tasks like emotion detection.

Across both phases, model performance was interpreted not only through metrics, but also via visual tools such as heatmaps, training curves, and word clouds. Model architecture decisions were grounded in empirical evidence, including text length distributions and resource constraints. All models were evaluated on the same stratified test split to ensure comparability.

Bringing these phases together, we demonstrated that while traditional models like SVM and Logistic Regression are effective baselines, deep learning especially CNNs provides significant advantages in capturing subtle emotional cues embedded in short sequences of text. The project thus offers both a validated solution and a methodological blueprint for future emotion classification work.