Comparative Evaluation of Classical Machine Learning Models for Sentiment Classification of Posts on X (Formerly Twitter)

Dr. Yogesh V Somwanshi[1], Mr. Dinesh Kumar Singh[2]
[1]*Department of Computer Science, Baliram Patil Arts, Commerce and Science College, Kinwat, Maharashtra, India*
[2]*Senior Lead Application Developer, GP Technology LLC, Client-The Federal Home Loan Mortgage Corporation*

***Abstract*—X (formerly Twitter) produces large volumes of short, informal text whose sentiment must be classified automatically for brand monitoring, public-opinion analysis and moderation. This paper evaluates three classical machine learning classifiers Multinomial Naive Bayes (MNB), Logistic Regression (LR) and a Linear Support Vector Machine (SVM) trained on TF-IDF word uni-gram and bi-gram features for three-class sentiment classification (negative, neutral, positive) on the TweetEval sentiment benchmark (45,615 training, 2,000 validation and 12,284 test posts). A social-media-specific preprocessing pipeline normalizes mentions, URLs, hashtags, emojis and elongated words. On the test set, LR achieved the highest macro-F1 (56.80%) and accuracy (58.17%), closely followed by the Linear SVM (56.54% macro-F1, 57.84% accuracy), while MNB obtained 54.36% macro-F1. MNB trained in about 0.01 s, the SVM in 1.21 s and LR in 5.71 s, and inference for all three took under 0.01 s per 1,000 posts. The results provide a reproducible, low-cost baseline for later comparison with neural and transformer-based models.**

***Index Terms*—Sentiment analysis, text classification, machine learning, X (Twitter), TF-IDF, logistic regression, support vector machine, Naive Bayes, TweetEval.**

I. INTRODUCTION

Social media platforms have become a primary channel for public expression. On X, users publish short posts about products, politics, health and current events. Automatically assigning such posts to categories, and in particular determining their sentiment, is a core natural language processing (NLP) task that supports brand monitoring, crisis response and content moderation [1], [2].

Classifying X posts is harder than classifying formal documents. Posts are short, contain misspellings, abbreviations, emojis, hashtags, user mentions and URLs, and often depend on context that is not present in the text [3]. Label distributions are typically imbalanced, and language use changes quickly over time [4].

Early approaches used bag-of-words features with Support Vector Machines and Naive Bayes [5], [11]. Neural models [6], [7] and, more recently, pre-trained transformers such as BERT [8] and BERTweet [15] represent the state of the art, but they require substantially more computation and engineering effort. Well-tuned linear models on TF-IDF features remain important because they are fast, small and easy to deploy, and they provide the baseline against which more complex models must be judged.

This paper therefore provides a controlled evaluation of three classical classifiers on a public X sentiment benchmark. The contributions are:

1) A reproducible preprocessing and TF-IDF classification pipeline tailored to X posts.
2) Test-set results for MNB, LR and Linear SVM on the TweetEval sentiment task, reported with accuracy, precision, recall and macro-F1.
3) A comparison of predictive quality against training time and model size.
4) A discussion of limitations and of the extension to neural and transformer-based models.

The remainder of the paper is organized as follows. Section II reviews related work. Section III describes the methodology. Section IV details the experimental

setup. Section V presents and discusses the results. Section VI covers limitations and ethical considerations, and Section VII concludes.

II. RELATED WORK

A. Classical approaches. Joachims showed that SVMs with high-dimensional sparse features are effective for text categorization [5]. Go et al. used distant supervision with emoticon-labelled tweets to train sentiment classifiers, an early influential study on Twitter data [11]. Shared tasks such as SemEval-2017 Task 4 established benchmarks for Twitter sentiment analysis [12]. Pang and Lee surveyed the broader field of opinion mining and sentiment analysis [2].

B. Neural approaches. Kim applied convolutional neural networks over word embeddings for sentence classification [6]. Recurrent architectures, especially LSTM [7], capture sequential dependencies, and pre-trained embeddings such as GloVe [13] provide dense word representations.

C. Transformer-based approaches. The Transformer architecture [14] enabled large-scale pre-training. BERT [8] and RoBERTa [9] achieved strong results on many NLP tasks, while DistilBERT [10] offers a smaller and faster alternative. BERTweet [15] was pre-trained on English tweets and outperforms general-domain models on several tweet tasks. Loureiro et al. showed that language models need periodic updating to handle temporal change in social media language [4].

D. Benchmarks. TweetEval unifies seven tweet classification tasks, including sentiment, emotion, hate, irony, offensive language and stance, under one evaluation framework [3]. Its sentiment task is derived from SemEval-2017 Task 4 [12]. Davidson et al. studied the difficulty of separating hate speech from merely offensive language [16].

III. METHODOLOGY

The pipeline consists of four stages: data acquisition, preprocessing, feature representation and classification.

A. Dataset. The TweetEval sentiment task [3] is used. It contains posts labelled negative, neutral or positive. The official splits are used: 45,615 posts for training, 2,000 for validation and 12,284 for testing. The validation set is used only to select hyper-parameters, and all reported scores are computed on the test set.

B. Preprocessing. The following normalization steps are applied to every post:

1) URLs are replaced with the token HTTPURL and user mentions with @USER.
2) Hashtags are split into words at lower-to-upper case boundaries (e.g., #MachineLearning becomes Machine Learning).
3) Emojis are converted to textual descriptions.
4) Characters repeated four or more times are reduced to two (e.g., sooooo becomes soo), and whitespace is normalized.
5) The text is lower-cased.

C. Feature representation. Posts are represented by TF-IDF vectors over word uni-grams and bi-grams. Terms occurring in fewer than two training posts are discarded. Term frequency is sub-linearly scaled and the inverse document frequency is smoothed:

$$w(t,d) = (1 + \ln tf(t,d)) \cdot [\ln((1+N)/(1+df(t))) + 1] \quad (1)$$

where tf(t,d) is the count of term t in post d, df(t) is the number of training posts containing t, and N is the number of training posts. Each vector is L2-normalized.

D. Classification models

1) Multinomial Naive Bayes (MNB): a probabilistic baseline that assumes conditional independence of features given the class.
2) Logistic Regression (LR): a linear discriminative model with L2 regularization, trained for multi-class prediction.
3) Linear SVM: maximizes the margin between classes and performs well on sparse, high-dimensional data [5].

IV. EXPERIMENTAL SETUP

A. Implementation. All models are implemented with scikit-learn [17]. The vectorizer is fitted on the training posts only. Each experiment is a single run, and the three classifiers are deterministic given the training data.

B. Hyper-parameters. The settings in Table I were explored, and for each model the value giving the highest macro-F1 on the validation set was used for the final test evaluation.

Table I. Hyper-parameter search

Model	Settings searched
TF-IDF	uni- and bi-grams, min_df = 2, sub-linear tf
MNB	alpha in {0.1, 0.5, 1.0}
LR	C in {0.1, 1, 10}; L2 penalty; max 2000 iterations
Linear SVM	C in {0.1, 1, 10}

C. Evaluation metrics. Because the classes are not equally frequent, accuracy alone is insufficient. The following are computed on the test set, with precision, recall and F1 macro-averaged over the three classes:

- Accuracy = (TP + TN) / (TP + TN + FP + FN)
- Precision = TP / (TP + FP)
- Recall = TP / (TP + FN)
- F1 = 2 · Precision · Recall / (Precision + Recall)
- Macro-F1: the unweighted mean of per-class F1 scores.

In addition, the wall-clock training time of the final model, the inference time for the first 1,000 test posts, and the size of the serialized vectorizer and model are recorded to quantify computational cost.

V. RESULTS AND DISCUSSION

A. Results. Table II reports the test-set performance of the three classifiers, and Table III their computational cost. Fig. 1 plots the same performance metrics.

Table II. Test-set performance (%)

Model	Acc.	Prec.	Rec.	Macro-F1
MNB + TF-IDF	56.71	56.89	55.14	54.36
LR + TF-IDF	58.17	57.49	57.52	56.80
Linear SVM + TF-IDF	57.84	57.22	57.55	56.54

Table III. Computational cost

Model	Train time (s)	Latency / 1k posts (s)	Size (MB)
MNB + TF-IDF	0.01	< 0.01	6.88
LR + TF-IDF	5.71	< 0.01	4.58
Linear SVM + TF-IDF	1.21	< 0.01	4.58

Model size is the serialized vectorizer plus classifier. Latency was below the resolution of the timer (0.01 s).

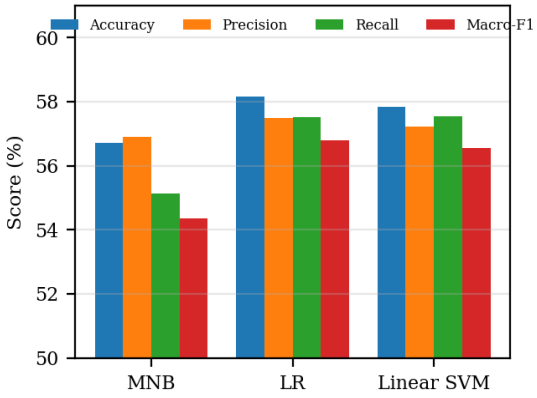

Fig. 1. Accuracy, precision, recall and macro-F1 of the three classifiers on the TweetEval sentiment test set.

B. Analysis. Overall performance. LR obtained the highest accuracy (58.17%) and macro-F1 (56.80%). The Linear SVM was very close, with macro-F1 only 0.26 points lower (56.54%) and a macro-averaged recall (57.55%) marginally above LR (57.52%). MNB was clearly weaker, 2.44 macro-F1 points below LR (Table II, Fig. 1). The official TweetEval metric for this task is macro-averaged recall [3], which corresponds to the Recall column. The absolute scores are modest, reflecting the difficulty of three-class sentiment classification of short posts with sparse lexical features; prior work suggests that contextual and tweet-specific models are needed to go substantially higher [3], [15].

Accuracy versus macro-F1. The gap between accuracy and macro-F1 was 2.35 points for MNB, 1.37 for LR and 1.30 for the SVM. The larger gap for MNB indicates that its predictions are less balanced across the classes than those of the discriminative models. Per-class results would be needed to identify which class is affected.

Training cost. MNB trained in about 0.01 s because it only accumulates class-conditional term counts. The

SVM needed 1.21 s and LR 5.71 s, about 4.7 times longer than the SVM (Table III, Fig. 2). The SVM therefore reached within 0.26 macro-F1 points of LR in about 21% of the training time, making it an attractive option when models must be retrained frequently.

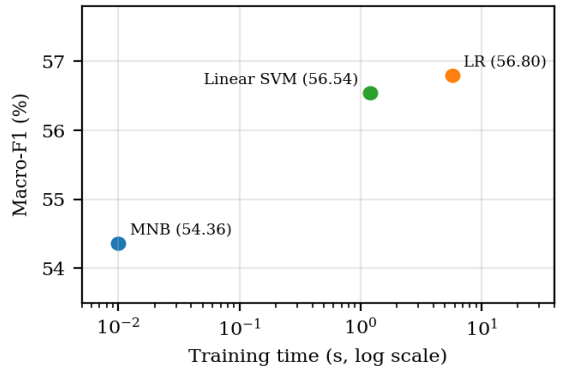

Fig. 2. Macro-F1 versus training time (log scale).

Model size and latency. All three pipelines are small. LR and the SVM each occupy 4.58 MB, while MNB occupies 6.88 MB because the serialized model stores a log-probability for every class and feature. Inference on 1,000 posts took less than 0.01 s for every model, so latency does not distinguish them in this setting (Fig. 3).

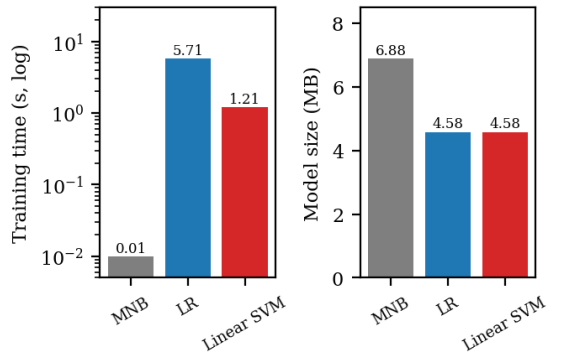

Fig. 3. Training time (left, log scale) and serialized model size (right).

Reliability of the differences. With 12,284 test posts, the standard error of an accuracy near 58% is about 0.45 points, i.e., roughly ±0.9 points at the 95% level. The 0.33-point accuracy difference between LR and the SVM is smaller than this, so the two should be regarded as comparable. The larger gaps between MNB and the other two models (1.46 and 1.13 accuracy points; 2.44 and 2.18 macro-F1 points) are more likely to be real, but a paired significance test such as McNemar's test on the predictions, which was not performed here, would be required to confirm them.

VI. LIMITATIONS AND ETHICAL CONSIDERATIONS

Limitations. (i) Only classical models were evaluated; recurrent and transformer-based models [7], [8], [10], [15] are not included, so no claim is made about their relative performance. (ii) Only one English sentiment task was used, and results may not transfer to other tasks or to code-mixed languages. (iii) Each result comes from a single run without a significance test. (iv) Latency was below timer resolution, so inference speed could not be compared. (v) Models trained on data from one period may degrade as language and topics evolve [4]. (vi) Crowd-sourced sentiment labels can be subjective, especially for sarcasm [16].

Ethics. Posts may contain personal information. The public benchmark used here is anonymized, and any application to new data should respect the platform's terms and applicable data-protection law. Automated sentiment systems can be biased against dialects or communities, so fairness audits and human review are recommended before deployment.

VII. CONCLUSION AND FUTURE WORK

This paper evaluated Multinomial Naive Bayes, Logistic Regression and a Linear SVM on TF-IDF features for three-class sentiment classification of X posts using the TweetEval benchmark. Logistic Regression gave the best macro-F1 (56.80%), the Linear SVM was nearly equal (56.54%) at roughly one fifth of the training time, and Naive Bayes was fastest but about 2.4 points lower (54.36%). These results form a fast and reproducible baseline. Future work will add a Bi-LSTM and fine-tuned transformer models such as DistilBERT and BERTweet, report per-class results and confusion matrices, apply paired significance tests over multiple seeds, and extend the study to other TweetEval tasks and to multilingual and code-mixed posts.

REFERENCES

[1] Aggarwal, C. C., & Zhai, C. (Eds.). (2012). Mining text data. Springer.

[2] Pang, B., & Lee, L. (2008). Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval, 2(1–2), 1–135.

[3] Barbieri, F., Camacho-Collados, J., Espinosa Anke, L., & Neves, L. (2020). TweetEval: Unified benchmark and comparative evaluation for tweet classification. In Findings of the Association for Computational Linguistics: EMNLP 2020 (pp. 1644–1650).

[4] Loureiro, D., Barbieri, F., Neves, L., Espinosa Anke, L., & Camacho-Collados, J. (2022). TimeLMs: Diachronic language models from Twitter. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations (pp. 251–260).

[5] Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. In Proceedings of the European Conference on Machine Learning (ECML) (pp. 137–142).

[6] Kim, Y. (2014). Convolutional neural networks for sentence classification. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP) (pp. 1746–1751).

[7] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735–1780.

[8] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of NAACL-HLT (pp. 4171–4186).

[9] Liu, Y., et al. (2019). RoBERTa: A robustly optimized BERT pretraining approach. arXiv preprint arXiv:1907.11692.

[10] Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108.

[11] Go, A., Bhayani, R., & Huang, L. (2009). Twitter sentiment classification using distant supervision. CS224N Project Report, Stanford University.

[12] Rosenthal, S., Farra, N., & Nakov, P. (2017). SemEval-2017 Task 4: Sentiment analysis in Twitter. In Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017) (pp. 502–518).

[13] Pennington, J., Socher, R., & Manning, C. D. (2014). GloVe: Global vectors for word representation. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP) (pp. 1532–1543).

[14] Vaswani, A., et al. (2017). Attention is all you need. In Advances in Neural Information Processing Systems (NeurIPS) (pp. 5998–6008).

[15] Nguyen, D. Q., Vu, T., & Nguyen, A. T. (2020). BERTweet: A pre-trained language model for English Tweets. In Proceedings of EMNLP: System Demonstrations (pp. 9–14).

[16] Davidson, T., Warmsley, D., Macy, M., & Weber, I. (2017). Automated hate speech detection and the problem of offensive language. In Proceedings of the International AAAI Conference on Web and social media (ICWSM) (pp. 512–515).

[17] Pedregosa, F., et al. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12, 2825–2830.