Depression Detection from Social Media Text: A Comprehensive Machine Learning Framework with Multi-Method Vectorization and Distribution Drift Analysis

Yiming Cheng[1] **Yi Wu**[1]
[1]Department of Computer Science, University of Chicago, MPCS Predoc
eaminchan@uchicago.edu, yiwu@uchicago.edu

Abstract

We present a streamlined depression-detection pipeline that concentrates on model training and evaluation for social media text. Starting from 10,325 depressed and 22,245 normal Weibo users, we aggregate each user's timeline, normalize the labels, and explore multiple linguistic representations (TF-IDF, N-grams, Word2Vec, and GloVe), both with and without PCA compression. Logistic regression, neural networks trained with stochastic gradient descent, and a k-means clustering baseline expose the trade-offs between sparse lexical counts and dense semantic embeddings under class imbalance and distribution drift. We evaluate every configuration with accuracy, precision, recall, F1-score, and AUC-ROC, demonstrating that Word2Vec/GloVe embeddings paired with the MLP consistently exceed F1 = 0.94 on balanced splits while remaining robust on the naturally imbalanced data. Code and processed data are publicly available: `https://github.com/EaminC/Math4ML/` and `https://drive.google.com/drive/folders/1w3JKnouiu-FNb2Qmk_QmMlcsFGuoVg-o?usp=sharing`.

1 Introduction and Motivation

Depression remains underdiagnosed despite its global prevalence World Health Organization [2023]. Social media platforms such as Weibo provide large-scale textual traces that encode linguistic and behavioral markers De Choudhury et al. [2013], making them attractive testbeds for automated screening. Yet practical deployments still wrestle with three recurring issues: (1) **class imbalance** between depressed and normal users, (2) **distribution drift** between training and evaluation cohorts Quiñonero-Candela et al. [2009], and (3) **feature representation** choices that dictate what linguistic signal is captured.

We tackle these challenges with a compact benchmarking framework that stresses classifier training rather than exploratory analysis. Contributions: (1) a reproducible user-level corpus that aggregates timelines into documents while exposing both the natural and balanced class ratios; (2) a systematic evaluation of TF-IDF, N-grams, Word2Vec, and GloVe embeddings, each with/without PCA compression; (3) comparative training of logistic regression, an MLP, and a label-free k-means baseline under identical splits to highlight distribution drift effects; and (4) comprehensive reporting of precision, recall, F1, accuracy, and AUC across all configurations with public code and processed data.

CMSC 25300 / 35300 (Spring 2025) Course Final Project

2 Dataset and Preprocessing

Our corpus covers 32,570 Weibo users (10,325 depressed, 22,245 normal). Two annotators apply a ten-criterion checklist rooted in clinical guidance; users with five or more positive indicators receive the depressed label, otherwise normal. This majority vote preserves the natural 1:2.15 prevalence, and we additionally release a balanced variant obtained by downsampling the normal class to 10,325 users.

For each user we concatenate the timeline into one document, normalize punctuation/emojis, drop non-Chinese tokens, and attach coarse engagement metadata (followers, followees, tweet counts, picture ratios). These document-level vectors feed every embedding method in Section 3.3. All experiments share a stratified five-fold split (fold 0 held out for testing) so that embeddings and classifiers only observe the remaining folds. Processed datasets, embeddings, and scripts accompany the release for reproducibility.

3 Proposed Methodology

Our pipeline addresses mental health text classification challenges through a systematic approach. **Data preprocessing**: We address class imbalance and distribution drift between training and test sets Quiñonero-Candela et al. [2009], quantified via Kullback-Leibler divergence. We employ SMOTE Chawla et al. [2002] or random undersampling for class balancing.

3.1 Text Vectorization

We concatenate each user's posts into a single document and compare four lightweight text representations. **TF-IDF** captures word salience with 5,000 unigram features after filtering extremely rare/common tokens. **N-grams** extend this idea with unigrams/bigrams/trigrams to encode short phrases that often signal emotion or self-disclosure. **Word2Vec** trains 100-dimensional CBOW embeddings (window=5, min count=2) on the corpus and averages each user's token vectors to obtain dense semantic document representations. **GloVe** follows the same averaging recipe but trains with a wider context window (10) to emphasize global co-occurrence structure. Full formulas and tokenization specifics appear in the appendix, allowing the main text to focus on cross-method comparisons rather than implementation minutiae.

3.2 Dimensionality Reduction with PCA

All sparse and dense embeddings are evaluated both in their original dimensionality and after Principal Component Analysis (PCA) compression. We retain the minimum number of components whose cumulative variance exceeds 95%, which trims TF-IDF/N-gram vectors from 5,000 features to roughly 2,000/500 components and compresses the dense Word2Vec/GloVe representations to about 50 dimensions. Scree plots and the full dimensionality table remain available in the appendix, while the main text focuses on how these compressed variants affect downstream metrics.

3.3 Classifiers

We evaluate three complementary approaches:

- **Logistic Regression**: a linear classifier that models $P(Y = 1|\mathbf{x})$ via a sigmoid over a weighted sum of embedding features. Its L2-regularized objective is

$$\min_{\mathbf{w},b} \sum_{i=1}^{n} \log\big(1 + \exp(-y_i(\mathbf{w}^\top \mathbf{x}_i + b))\big) + \frac{\lambda}{2}\|\mathbf{w}\|_2^2, \tag{1}$$

  with predictions $p(y = 1|\mathbf{x}) = \sigma(\mathbf{w}^\top \mathbf{x} + b)$.
- **Neural Networks (MLP)**: a single-hidden-layer feedforward network with ReLU activations and Adam optimization, capturing non-linear interactions:

$$\mathbf{h} = \sigma(\mathbf{W}_1\mathbf{x} + \mathbf{b}_1), \qquad \hat{y} = \sigma(\mathbf{w}_2^\top \mathbf{h} + b_2), \tag{2}$$

  trained via Adam updates $\theta_{t+1} = \theta_t - \eta\hat{m}_t/(\sqrt{\hat{v}_t} + \epsilon)$.

- **Unsupervised k-means**: a label-free clustering baseline that minimizes $\sum_i \min_j \|\mathbf{x}_i - \mathbf{c}_j\|_2^2$ with $k = 2$, maps each centroid to a class via majority vote, and uses soft membership weights for AUC computation.

3.4 Evaluation Metrics

We report F1-score, Precision, Recall, Accuracy, and Area Under the ROC Curve (AUC-ROC) on the held-out test fold. Definitions appear in Appendix A.5.

4 Experimental Evaluation

4.1 Train/Test Protocol

We create a stratified K-fold partition with $K = 5$ and designate one fold (fold 0 in our runs) as the held-out test set. The remaining 80% of the data serve as the sole training split for every embedding/model configuration. Balanced datasets are downsampled beforehand so both classes contain 10,325 users, ensuring each fold preserves a 50/50 ratio. The unbalanced dataset retains the natural 1:2.15 prevalence to evaluate robustness under real-world skew. For every vectorization method (TF-IDF, N-grams, Word2Vec, GloVe) and embedding variant (original/PCA), we train logistic regression, the MLP, and the unsupervised k-means classifier on the training portion only, then evaluate all metrics exclusively on the test fold.

4.2 Classification Performance

Tables 1 and 2 list the held-out test results for every dataset/method/model combination using original embeddings and PCA-compressed embeddings respectively. Dense semantic representations (Word2Vec and GloVe) paired with the MLP achieve the strongest F1 and AUC scores on both balanced and unbalanced splits. High-dimensional sparse representations trail by roughly 10 percentage points in F1, and PCA compression hurts TF-IDF/N-grams more noticeably than the dense embeddings. The unsupervised k-means baseline highlights the performance gap relative to supervised models, especially on sparse features where clusters poorly align with depression labels.

Across all three classifiers, the semantic embeddings consistently outperform the sparse bag-of-words features. For example, on the balanced dataset with original embeddings, the best TF-IDF configuration reaches $F_1 = 0.832$ (MLP) and AUC $= 0.915$, whereas Word2Vec and GloVe each exceed $F_1 = 0.94$ and $AUC = 0.98$ regardless of whether we use logistic regression, MLP, or even the k-means baseline. The same pattern holds on the unbalanced split: Word2Vec/GloVe maintain $F_1 \approx 0.92$ with AUC above 0.97, while TF-IDF and N-grams drop to $F_1 \leq 0.77$ under logistic regression and degrade further under k-means. These results confirm that dense embeddings provide richer, more robust signal for depression detection than purely lexical counts, even when the downstream model is as simple as logistic regression.

Embedding separability via t-SNE. To interpret these gains, we project 500 balanced samples per class into two dimensions with t-SNE (Figure 1). TF-IDF and N-gram embeddings produce interleaved clusters with no clear margin, consistent with their lower test metrics. In contrast, Word2Vec and GloVe yield two compact groups with minimal overlap, reflecting the semantic structure captured by dense embeddings and foreshadowing their superior performance across all classifiers.

The gap between balanced and unbalanced performance highlights the expected precision-recall trade-off when models are exposed to the natural 1:2.15 class ratio. Dense embeddings still maintain high recall (≈ 0.90), suggesting that linguistic signals overcome class imbalance when the model is expressive enough. Conversely, sparse representations rely on logistic regression to prevent false positives, underscoring the need for additional sampling strategies (e.g., SMOTE) if they are to match the dense embedding results.

Table 1: Held-out test results using original embeddings.

Dataset	Vectorization	Model	Accuracy	Precision	Recall	F1	AUC
Balanced	TF-IDF	LogReg	0.832	0.841	0.820	0.830	0.914
		MLP	0.833	0.835	0.830	0.832	0.915
		KMeans	0.519	1.000	0.039	0.075	0.718
	N-grams	LogReg	0.827	0.826	0.829	0.827	0.909
		MLP	0.837	0.827	0.852	0.839	0.912
		KMeans	0.500	0.500	1.000	0.667	0.503
	Word2Vec	LogReg	0.937	0.943	0.930	0.937	0.979
		MLP	0.948	0.960	0.935	0.947	0.985
		KMeans	0.774	0.717	0.905	0.800	0.879
	GloVe	LogReg	0.943	0.950	0.935	0.942	0.980
		MLP	0.952	0.965	0.938	0.951	0.986
		KMeans	0.792	0.734	0.917	0.815	0.893
Unbalanced	TF-IDF	LogReg	0.846	0.739	0.795	0.766	0.918
		MLP	0.861	0.890	0.642	0.746	0.917
		KMeans	0.683	0.000	0.000	0.000	0.500
	N-grams	LogReg	0.828	0.689	0.835	0.755	0.907
		MLP	0.859	0.899	0.625	0.737	0.911
		KMeans	0.683	0.000	0.000	0.000	0.500
	Word2Vec	LogReg	0.939	0.888	0.924	0.906	0.977
		MLP	0.953	0.944	0.906	0.925	0.983
		KMeans	0.694	0.510	0.924	0.657	0.872
	GloVe	LogReg	0.939	0.890	0.922	0.906	0.978
		MLP	0.951	0.940	0.903	0.921	0.982
		KMeans	0.717	0.531	0.931	0.676	0.886

Table 2: Held-out test results using PCA-compressed embeddings.

Dataset	Vectorization	Model	Accuracy	Precision	Recall	F1	AUC
Balanced	TF-IDF	LogReg	0.798	0.839	0.738	0.785	0.871
		MLP	0.799	0.798	0.801	0.799	0.871
		KMeans	0.533	1.000	0.067	0.125	0.871
	N-grams	LogReg	0.776	0.734	0.865	0.794	0.864
		MLP	0.808	0.791	0.836	0.813	0.884
		KMeans	0.500	0.500	1.000	0.667	0.500
	Word2Vec	LogReg	0.935	0.940	0.928	0.934	0.977
		MLP	0.947	0.953	0.940	0.947	0.985
		KMeans	0.771	0.714	0.905	0.798	0.879
	GloVe	LogReg	0.938	0.945	0.931	0.938	0.978
		MLP	0.942	0.947	0.936	0.941	0.982
		KMeans	0.791	0.733	0.917	0.814	0.893
Unbalanced	TF-IDF	LogReg	0.838	0.756	0.723	0.739	0.876
		MLP	0.855	0.835	0.677	0.748	0.883
		KMeans	0.683	0.000	0.000	0.000	0.500
	N-grams	LogReg	0.739	0.557	0.866	0.678	0.861
		MLP	0.832	0.847	0.576	0.686	0.881
		KMeans	0.683	0.000	0.000	0.000	0.500
	Word2Vec	LogReg	0.935	0.880	0.919	0.899	0.975
		MLP	0.951	0.944	0.899	0.921	0.982
		KMeans	0.694	0.509	0.924	0.657	0.873
	GloVe	LogReg	0.938	0.888	0.922	0.905	0.977
		MLP	0.951	0.940	0.905	0.922	0.983
		KMeans	0.714	0.528	0.931	0.674	0.886

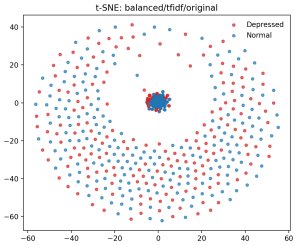
5 Expected Contributions

In summary, we showcase how classical ML ingredients (vectorizers, PCA, gradient-based classifiers) behave under class imbalance and distribution drift for depression detection, and we release all code plus processed datasets to enable exact replication: `https://github.com/EaminC/Math4ML/` and `https://drive.google.com/drive/folders/1w3JKnouiu-FNb2Qmk_QmMlcsFGuoVg-o?usp=sharing`.

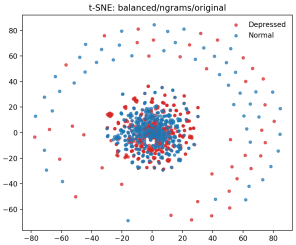
References

Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

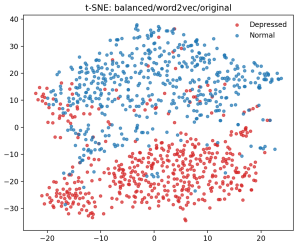
Munmun De Choudhury, Michael Gamon, Scott Counts, and Eric Horvitz. Predicting depression via social media. In *Proceedings of the international AAAI conference on web and social media*, volume 7, pages 128–137, 2013.



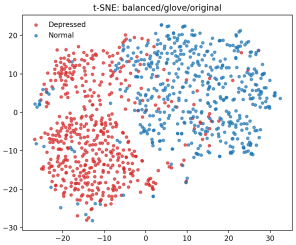
(a) TF-IDF (balanced)



(b) N-grams (balanced)



(c) Word2Vec (balanced)



(d) GloVe (balanced)

Figure 1: t-SNE visualizations of balanced dataset embeddings. Word2Vec and GloVe show well-separated clusters, whereas TF-IDF and N-grams exhibit heavier overlap and noisier structure.

Joaquin Quiñonero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. *Dataset shift in machine learning*. MIT Press, 2009.

World Health Organization. Depression, 2023. URL `https://www.who.int/news-room/fact-sheets/detail/depression`. Accessed: 2024.

A Mathematical Formulations

A.1 Distribution Drift Quantification

We quantify distribution drift using Kullback-Leibler divergence:

$$D_{\mathrm{KL}}(P_{\mathrm{test}} \| P_{\mathrm{train}}) = \sum_{x,y} P_{\mathrm{test}}(x,y) \log \frac{P_{\mathrm{test}}(x,y)}{P_{\mathrm{train}}(x,y)} \tag{3}$$

where $P_{\mathrm{train}}(X,Y)$ and $P_{\mathrm{test}}(X,Y)$ denote the training and test distributions, respectively.

A.2 Text Vectorization

TF-IDF: For a term t in document d, the TF-IDF score is:

$$\text{TF-IDF}(t,d) = \text{TF}(t,d) \times \log \frac{N}{df(t)} \tag{4}$$

where N is the total number of documents and $df(t)$ is the document frequency of term t.

Word2Vec: Word2Vec learns word representations by maximizing the log-likelihood:

$$\sum_{t=1}^{T} \sum_{-c \le j \le c, j \neq 0} \log p(w_{t+j} | w_t) \tag{5}$$

where c is the context window size and T is the sequence length.

A.3 Dimensionality Reduction

For a data matrix $X \in \mathbb{R}^{n \times p}$, PCA finds the principal components by solving:

$$\max_{\mathbf{w}} \mathbf{w}^T \mathbf{\Sigma} \mathbf{w} \quad \text{subject to} \quad \|\mathbf{w}\|_2 = 1 \tag{6}$$

where $\mathbf{\Sigma}$ is the covariance matrix.

A.4 Classification Algorithms

Logistic Regression: Models the probability $P(Y = 1|X)$ using:

$$P(Y = 1|X) = \frac{1}{1 + \exp(-\mathbf{w}^T \mathbf{x} - b)} \tag{7}$$

Support Vector Machines: Find the optimal hyperplane by solving:

$$\min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^{n} \xi_i \tag{8}$$

subject to $y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \xi_i$ for all i.

Neural Networks with Gradient Descent: We train feedforward networks using stochastic gradient descent (SGD) with updates:

$$\theta_{t+1} = \theta_t - \eta \nabla_\theta \mathcal{L}(\theta_t; x_i, y_i) \tag{9}$$

where η is the learning rate and $\mathcal{L}$ is the loss function.

A.5 Evaluation Metrics

We report comprehensive metrics to evaluate classification performance:

$$\text{F1-score:} \quad F_1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{10}$$

$$\text{Precision:} \quad P = \frac{TP}{TP + FP} \tag{11}$$

$$\text{Recall:} \quad R = \frac{TP}{TP + FN} \tag{12}$$

$$\text{Accuracy:} \quad A = \frac{TP + TN}{TP + TN + FP + FN} \tag{13}$$

where TP, TN, FP, and FN denote true positives, true negatives, false positives, and false negatives, respectively. The Area Under the ROC Curve (AUC-ROC) measures the classifier's ability to distinguish between classes across all possible classification thresholds, providing a threshold-independent performance metric.