Classical Machine Learning Models vs Deep Learning Approaches for Text Classification Tasks

Alyssa Ting, Freya Gulamali, Jaeden Toy, Jessica Oldov, Kartik Pejavara, Yulia Savine

Abstract

This study aims to compare classical machine learning (ML) models (random decision forest, support vector machine (SVM), K-nearest neighbors (KNN)) to deep learning (DL) models (BERT) for text classification. We compared models by training and testing each model on the Stanford Sentiment Treebank, and evaluating performance using accuracy, recall, precision, and F1-score. BERT performed better than the other models on all metrics, but this required greater computational resources. Use of classical versus DL and ML models for classification depends on the nature of the task.

1. Problem Statement

NLP models are rapidly advancing. However, this often comes at the increased complexity of the model, creating compute requirement barriers to their use. In this study, we compare various simple, classical ML models to a DL model in a classification task to determine if the DL model indeed has better performance, and further, if this better performance is worth the increased complexity.

2. Background & Related Works

Prior studies in text classification have compared classical models [1], [2], [3], [4]. Others attempted to develop novel approaches using a hybrid of model types [5].

2.1 Classical Models and Text Multiclass Classification

Shah et al. compared the accuracy of logistic regression, random forest, and KNN models in text multiclass classification using a BBC dataset with five categories. After vectorizing the data using term frequency-inverse document frequency, the logistic regression model demonstrated the greatest accuracy across the classification methods. Potential limitations included the class imbalance, large feature space generated by text classification. The authors also recognized that for more complex datasets, part-of-speech tagging and image recognition could play a role in classification. While logistic regression demonstrated higher accuracy over other methods, it was highlighted that random forests have been found to perform well in real world scenarios beyond the scope of this research [6].

2.2 Classical Models vs Deep Learning Approaches for Computer Vision

In their study, Karypidis et al. investigated traditional ML and DL methods for 2D object classification using the Belgium Traffic Sign Dataset. Traditional ML methods such as Bag of Visual Words (BOVW), KNN, and SVM were tested alongside DL techniques including the VGG16 architecture and a custom DCNN. Findings revealed that in traditional ML methods Manhattan distance outperforms Euclidean distance in the BOVW model, SVM accuracy fluctuates with vocab size, and KNN accuracy trends upwards with vocab size at the expense of increased computational complexity, indicating the importance of hyperparameter selection. DL methods, specifically pretrained VGG16 and the proposed DCNN, had similar performance, both surpassing traditional ML methods in accuracy. The DCNN, however, offered higher accuracy with lower computational complexity. Overall, Karypidis et al.

underscored the superiority of DL approaches in object classification within computer vision [7].

2.3 Classical Models vs Deep Learning Approaches for Binary Text Classification

Kamath et al. compared the accuracy of classical ML models (random forest classifier, logistic regression, SVM, naive bayes and a multilayer perceptron) to a DL (CNN) model for document classification. The authors used two datasets, each with multiple classes: Health, a dataset with images of insurance invoices belonging to 18 different classes, and Tobacco-3482, a dataset with images of tobacco belonging to nine different classes.

The authors vectorized the text documents, trained the five different models, and tested the models on both raw and processed documents. Then, they trained a CNN on the same raw and processed data. They found that while the logistic regression classifier performed the best of all the classical ML methods used, the CNN performed the best overall. The logistic regression classifier achieved its best accuracy of 81% of the raw Health dataset, while the CNN achieved its lowest accuracy of 82% on the raw Tobacco dataset and highest accuracy of 96% of the processed Tobacco dataset [8].

3. Description of Methods

In this paper, we compare the performance of three classical machine learning methods (random forests, SVM and KNN) to the performance of a DL model (BERT) on a text classification task.

3.1 Data Processing

3.1.1 Dataset

We train and evaluate the models on a dataset from the General Language Understanding Evaluation (GLUE) benchmark known as the Stanford Sentiment Treebank (SST2) [19]. SST2 consists of sentences from publicly available movie reviews, where each sentence is labeled with either “positive” or “negative” sentiment. The training set consisted of 57,246 sentences (55.6% positive, 44.4% negative) and the test set consisted of 10,103 sentences (56.7% positive, 43.3% negative). The data was split using a 85-15 train-test split (seed 33).

3.1.2 Processing and Vectorization

In order to determine which of the models performs best, it was necessary to process the data in the same way for accurate comparisons. We used the HuggingFace AutoTokenizer library to tokenize the SST2 dataset. Subsequently, the tokenized vectors were made into meaningful embeddings using the AutoModel library. This produced vectors of dimension 768, which were used to train the classical ML models. BERT was trained on its respective tokenizer and embedding models, as it was pre-trained on these.

3.2 Classical Models

After vectorizing data, we trained each of the classical models selected on this data. For each, we performed K-Fold Cross Validation to determine the optimal hyperparameter.

In the K-Fold Cross Validation, we split the data into three folds. For each hyperparameter value, we used these three folds to calculate the average performance of a classifier using that hyperparameter value. After finding the hyperparameter value with the best average performance on the three folds, we use that hyperparameter value to train the classifier and test it.

3.2.1 Random Decision Forest

A decision forest is an aggregate of decision trees that aims to reduce the prediction variance that is faced with decision trees (since single trees are arbitrarily complex). A decision tree is an optimal set of questions to classify samples that starts at a root and then splits the dataset across a new feature per level until the classification criteria is met. Then, the mean or mode of the leaf nodes is taken to make predictions [8].

For K-Fold Cross Validation, we tested hyperparameter values ranging from 50 to 300 (inclusive) trees, in increments of 25 due to

computational limitations. The best hyperparameter was 275 trees.

3.2.2 Support Vector Machine

Support Vector Machine (SVM) is a machine learning technique for binary classification, aiming to separate data into two groups. It accomplishes this by finding a hyperplane that maximizes the distance between the two groups, known as the margin. The points closest to this hyperplane, termed support vectors, are crucial in defining its position. In general, SVMs are effective in high-dimensional spaces and are commonly applied in various fields [9], [10], [11].

Due to computational limitations, it was not feasible to execute K-Fold Cross Validation on this classifier. We used the default parameters for the SVM classifier implemented by the Sci-kit Learn library instead.

3.2.3 K-Nearest Neighbors

K-nearest neighbors (KNN) is a classical machine learning method that has almost no overhead in training time, but at inference time, tends to be more computationally expensive than other methods. KNN classifies a data point by using a summarization function which is applied to the K-nearest training points to it, using many similarity metrics but often using Euclidean distance. For binary classification, the summarization function used is often the majority– simply returning the majority class of the group of K-nearest training points to the data point in question. This implies that for the same data point, a different value of K may yield a different classification for that data point. KNN is an algorithm frequently used for different types of classifications, and as a result, many different variations of the algorithm have been created to maximize its performance on a specified task [12], [13], [14], [15].

For K-Fold Cross Validation, we tested hyperparameter values ranging from 1 to 10 (inclusive) neighbors. The best hyperparameter we found was 1 nearest neighbor.

3.3 Deep Learning Model: Bidirectional Encoder Representations from Transformers (BERT)

BERT is a pre-trained transformer-based model. First introduced in [16], the transformer model achieved state-of-the-art performance on the machine translation task. Specifically, its self-attention mechanism captures relationships between distant elements in an input sequence, allowing it to understand complex patterns and dependencies. BERT is an encoder-only model that learns bi-directional representations of input sequences to improve contextual understanding [17]. Originally trained for next sentence prediction and masked language modeling, BERT presents an exceptional ability to understand text. Because of its language-understanding capabilities, it is often the de facto model for a variety of NLP tasks, including text classification [18].

In this study, we employ BERT as our DL model. Specifically, we fine-tune BERT-base-uncased, the smaller version of the BERT model trained on uncased input text. Using a 16GB TPU, we fine-tune for 3 epochs and use a learning rate of 2e-5, batch size of 16, and weight decay of 0.1. To ensure we use the best model, we implement model evaluation and checkpointing every one-tenth of the total training steps. The model with the lowest evaluation loss was selected for testing in the results. All fine-tuning and inference was done using the HuggingFace Transformers library.

3.5 Evaluation Metrics

Accuracy, precision, recall, and F1 score, are computed for each model. These metrics will be compared to arrive at final conclusions about the efficacy of classical machine learning methods as compared to deep learning methods when it comes to binary text classification.

4. Results

Table 1: Model performance results on the test set.

Model	Accuracy	Precision	Recall	F1 score
KNN	0.885	0.898	0.899	0.899
Random Forest	0.868	0.886	0.881	0.883
SVM	0.869	0.890	0.879	0.884
BERT	0.950	0.952	0.961	0.956

4.1 Classical ML Models

Our results are presented in Table 1. The traditional models performed similarly to each other. The accuracy scores were 0.885 for KNN, 0.868 for Random Forest, and 0.869 for SVM. The F1 scores were 0.899 for KNN, 0.883 for Random Forest, and 0.884 for SVM. The F1 score is an important indicator of performance because it takes into account both false positives and false negatives. Especially in cases of class imbalance as was the case with our test set containing more positives than negatives, F1 score can be more valuable than accuracy. The KNN performed slightly better than the Random Forest and SVM models according to accuracy, precision, recall, and F1 score metrics.

4.2 BERT

BERT achieved its lowest loss (0.203) after three-fifths of the total training steps (6444). The model at this checkpoint was used for testing. The results in Table 1 show a relatively high accuracy (0.950), indicating that the model was correct on its predictions a large majority of the time. The model also achieved high precision (0.952), recall (0.961) and F1 score (0.956), indicating that the model was intentional in its predictions and did not just assign confusing inputs to the class that made up the largest proportion of the dataset (positive). We break down the model results even further.

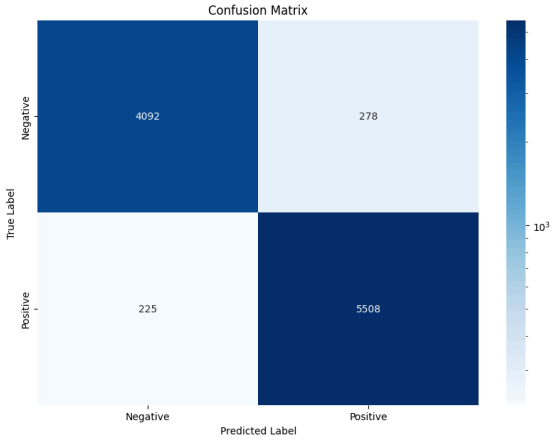

Fig. 1. BERT Confusion Matrix

From the confusion matrix, we can see that, even though the test set contained significantly more positive messages, the model misclassified positive texts (5.05% of the time) at a similar rate as it misclassified negative texts (5.50% of the time).

5. Discussion

5.1.1 Discussion of Results

From the performance results in Table 1, it is clear that BERT was the best model for this binary text classification task. The three classical models achieved relatively similar performance statistics. The random forest and SVM models had the most similar metrics, where the greatest difference was only 0.004 in precision. The KNN model achieved slightly better metrics than the other two classical models, usually an increase of 0.01 in each category. BERT, however, achieved a significantly higher accuracy, precision, recall, and F1 score than all of the classical models. It

was much better at correctly predicting a messages class than any of the other models. BERT also took 1.5 hours to train, which was more than the traditional models.

These findings are similar to previous work of Kamath et al. and Karypidis et al. who compared CNN performance to that of traditional models, finding superior performance of the CNN model. It is reasonable that a DL model like BERT, designed for language processing due to its bi-directional nature, would also perform better than the traditional models.

The KNN, SVM, and Random Forest models still performed decently in this use case of semantic analysis likely because there was limited class imbalance. The KNN performance may have been limited by the fact that the model had a nearest neighbor parameter of one. This means that there was potential overfitting and low variance, which would limit generalizability when the model was applied to the test set. The SVM may be limited by not being able to cross validate, which is a technique that also limits overfitting, due to high computational resources. Although cross validation led to the determination for 275 trees during Random Forest training, a high number of trees can also lead to overfitting and increase the time required to train.

Considering the time and computational resources required to train BERT, it may be worth considering use of traditional models in cases with minimal class imbalance and when the risk of inaccuracy is low. The traditional models still performed with an F1-score of about 0.883, 0.073 points lower than BERT.

5.1.2 Limitations

We faced several significant limitations related to the computational resources required for our methodologies. Firstly, the scalability of SVMs posed a major challenge. SVMs, especially with non-linear kernels, are known to scale poorly with large datasets, rendering the training process computationally expensive and time-consuming. The issue was further compounded by the memory constraints encountered during the vectorization of our original dataset, which included lengthy reviews. The process not only escalated the amount of RAM needed but also increased computational demands significantly. As a result, we ended up selecting an alternative dataset.

Furthermore, the computational requirements of training and testing SVMs impacted our ability to perform thorough hyperparameter tuning. The time it took to train and test the SVM (which exceeded an hour) made it computationally intractable for us to implement K-fold cross-validation. This limitation likely prevented us from identifying the optimal hyperparameters, which potentially affected the accuracy of our SVM classifier. Similarly, our limited resources restricted our ability to exhaustively test the range of hyperparameters for the Random Forest classifier. Opting for a larger step size in testing likely caused us to miss finer adjustments in hyperparameters that could have enhanced classifier performance.

6. Conclusion

Though complex models can have higher accuracy, this comes with tradeoffs. Future investigation involves determining tasks that would be better suited for traditional machine learning versus deep learning models depending on resources and accuracy. This involves understanding the financial and environmental impact associated with computational resources. This cost is sometimes not considered a major limitation of deep learning methods, especially in light of the popularization of large language models and other complex techniques. However, it is important to consider that the difference in accuracy may not be worth this investment. Another benefit to traditional models is interpretability, increasing trustworthiness of results. Our investigation into text classification is a preliminary step in adding to the literature of when this tradeoff is appropriate.

7. References

[1] S. Aseervatham, A. Antoniadis, E. Gaussier, M. Burlet, and Y. Denneulin, "A sparse version of the ridge logistic regression for large-scale text categorization," *Pattern Recognition Letters*, vol. 32, no. 2, pp. 101–106, Jan. 2011, doi: 10.1016/j.patrec.2010.09.023.

[2] R. Kumari and S. Srivastava, "Machine Learning: A Review on Binary Classification," *International Journal of Computer Applications*, vol. 160, pp. 11–15, Feb. 2017, doi: 10.5120/ijca2017913083.

[3] M. A. Kumar and M. Gopal, "A comparison study on multiple binary-class SVM methods for unilabel text categorization," *Pattern Recognition Letters*, vol. 31, no. 11, pp. 1437–1444, Aug. 2010, doi: 10.1016/j.patrec.2010.02.015.

[4] F. Colas and P. Brazdil, "On the Behavior of SVM and Some Older Algorithms in Binary Text Classification Tasks," in *Text, Speech and Dialogue*, P. Sojka, I. Kopeček, and K. Pala, Eds., Berlin, Heidelberg: Springer, 2006, pp. 45–52. doi: 10.1007/11846406_6.

[5] Y. Al Amrani, M. Lazaar, and K. E. El Kadiri, "Random Forest and Support Vector Machine based Hybrid Approach to Sentiment Analysis," *Procedia Computer Science*, vol. 127, pp. 511–520, Jan. 2018, doi: 10.1016/j.procs.2018.01.150.

[6] K. Shah, H. Patel, D. Sanghvi, and M. Shah, "A Comparative Analysis of Logistic Regression, Random Forest and KNN Models for the Text Classification," *Augment Hum Res*, vol. 5, no. 1, p. 12, Mar. 2020, doi: 10.1007/s41133-020-00032-0.

[7] E. Karypidis, S. G. Mouslech, K. Skoulariki, and A. Gazis, "Comparison Analysis of Traditional Machine Learning and Deep Learning Techniques for Data and Image Classification," *WSEAS TRANSACTIONS ON MATHEMATICS*, vol. 21, pp. 122–130, Mar. 2022, doi: 10.37394/23206.2022.21.19.

[8] C. N. Kamath, S. S. Bukhari, and A. Dengel, "Comparative Study between Traditional Machine Learning and Deep Learning Approaches for Text Classification," in *Proceedings of the ACM Symposium on Document Engineering 2018*, in DocEng '18. New York, NY, USA: Association for Computing Machinery, Aug. 2018, pp. 1–11. doi: 10.1145/3209280.3209526.

[9] J. Sujanaa, S. Palanivel, and M. Balasubramanian, "Emotion recognition using support vector machine and one-dimensional convolutional neural network," *Multimed Tools Appl*, vol. 80, no. 18, pp. 27171–27185, Jul. 2021, doi: 10.1007/s11042-021-11041-5.

[10] S. Goyal, "Effective software defect prediction using support vector machines (SVMs)," *Int J Syst Assur Eng Manag*, vol. 13, no. 2, pp. 681–696, Apr. 2022, doi: 10.1007/s13198-021-01326-1.

[11] V. Patil, M. Madgi, and A. Kiran, "Early prediction of Alzheimer's disease using convolutional neural network: a review," *The Egyptian Journal of Neurology, Psychiatry and Neurosurgery*, vol. 58, no. 1, p. 130, Nov. 2022, doi: 10.1186/s41983-022-00571-w.

[12] S. Uddin, I. Haque, H. Lu, M. A. Moni, and E. Gide, "Comparative performance analysis of K-nearest neighbour (KNN) algorithm and its different variants for disease prediction," *Sci Rep*, vol. 12, no. 1, p. 6256, Apr. 2022, doi: 10.1038/s41598-022-10358-x.

[13] H. Gweon, M. Schonlau, and S. H. Steiner, "The k conditional nearest neighbor algorithm for classification and class probability estimation," *PeerJ Comput Sci*, vol. 5, p. e194, May 2019, doi: 10.7717/peerj-cs.194.

[14] C. Feng, B. Zhao, X. Zhou, X. Ding, and Z. Shan, "An Enhanced Quantum K-Nearest Neighbor Classification Algorithm Based on Polar Distance," *Entropy*, vol. 25, no. 1, Art. no. 1, Jan. 2023, doi: 10.3390/e25010127.

[15] E. Ozturk Kiyak, B. Ghasemkhani, and D. Birant, "High-Level K-Nearest Neighbors (HLKNN): A Supervised Machine Learning Model for Classification Analysis," *Electronics*, vol. 12, no. 18, Art. no. 18, Jan. 2023, doi: 10.3390/electronics12183828.

[16] A. Vaswani et al., "Attention Is All You Need." arXiv, Aug. 01, 2023. doi: 10.48550/arXiv.1706.03762.

[17] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.” arXiv, May 24, 2019. doi: 10.48550/arXiv.1810.04805.

[18] B. Sabiri, A. Khtira, B. Asri, and M. Rhanoui, “Analyzing BERT’s Performance Compared to Traditional Text Classification Models,” Jan. 2023, pp. 572–582. doi: 10.5220/0011983100003467.

[19] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. Manning, A. Ng and C. Potts, “Recursive deep models for semantic compositionality over a sentiment treebank.” 2013. *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1631–1642.