OXFORD JOURNAL
OF STUDENT
SCHOLARSHIP

New York Times Headlines Sentiment Analysis: Classification, Regression, and Transformer Models

Pari Bhandari
paribhandari200906@gmail.com

ABSTRACT

Sentiment analysis is an important component of natural language processing, intending to determine the emotional tone of a text. This study investigates whether fine-tuned transformers outperform traditional machine learning models (classification and ordinal regression) on lexicon-labeled sentiment headlines with a severe class imbalance. The first section uses frozen BERT embeddings as input features for classification and ordinal regression models such as Random Forest, KNeighbors, Support Vector Machine, and Linear Regression, with hyperparameters optimized via GridSearchCV using 5-fold cross-validation on the training set. The second section directly employs a text-to-text transformer T5-small model evaluated in zero-shot and fine-tuned settings. Contrary to expectations, results demonstrate that Random Forest Classifier achieves the highest test macro F1 performance over fine-tuned T5-small and all regression models. While KNeighbors excelled in minority classes performance despite overfitting, all model families generally struggled to generalize across minority sentiment classes due to dataset imbalance. Fine-tuned T5-small, particularly at 5 epochs, achieves balanced performance across all sentiment labels, but collapses under premature early stopping. These findings challenge transformer superiority for short-text sentiment analysis, suggesting that BERT embeddings and stratified optimization are essential for achieving competitive performance on imbalance datasets. Further improvements could be achieved through larger models, hybrid embedding-transformer approaches, optimized hyperparameters, and balanced datasets.

INTRODUCTION

Sentiment analysis is a core component of natural language processing, concerned with deriving the emotional tone communicated by a piece of text. It is the process of gathering and analyzing people's opinions, thoughts, and impressions regarding various topics, products, subjects, and services (1). In practice, the sentiment of a text is either classified as positive, negative, or neutral or assigned a number from a range, denoting the degree of positivity or negativity. Also known as opinion mining, sentiment analysis has been a field of research for decades due to a variety of its applications in nearly every domain. It is frequently used in commercial decision-making that heavily relies on public opinion, as it can guide product development and marketing procedures. Moreover, with the advent of social media, there has been a massive increase in the amount of opinionated data available for analysis, pushing forward research in this area (2).

An important intersection of these ideas can be seen in news headlines. In today's digital age, news articles are more accessible than ever across online platforms, serving as the first point of contact between the media and the public. This accessibility significantly increases their reach and influence, allowing them to rapidly shape public opinion on a wide range of topics. However, most people obtain information by scanning news headlines instead of reading full articles (3), which is why the sentiment of the headline might change the way a piece of information is perceived. Despite their importance, it is difficult for models to understand their sentiment because of the short length or limited context of headlines. Previous research on headline sentiment focuses on discrete classification using simple transformer models (4). To build upon this, we not only use traditional machine learning models for both classification and ordinal regression but also implement an encoder-decoder transformer architecture, both of which undergo hyperparameter tuning.

This study evaluates the performance of a combination of BERT encoder (5) and optimized traditional machine learning models (both classification and ordinal regression) against a text-to-text transformer model for predicting TextBlob lexicon-based sentiment labels on imbalanced New York Times headlines (6). Specifically, sentence embeddings as extracted from the pre-trained BERT model are used for classification and regression purposes by multiple machine learning models, optimized through stratified train-test splitting and hyperparameter tuning. Then, a systematic comparison is drawn with the T5-small model (7) applied in both zero-shot and fine-tuned settings. This enables us to determine the effectiveness of one method over the other relative to their computational cost, in the context of headline sentiment analysis.

METHODS AND METRIALS

The dataframe for this study was sourced from the publicly available Kaggle repository and consists of New York Times headlines collected over a one week period in July 2024 (8).
The headline text was extracted from the Title column and used as the input variable, whereas the sentiment label was extracted from the Sentiment column and used as the target variable.
Additionally, the balance of the sentiment classes in the dataset was assessed using a counter. The dataset was found to be unbalanced, with a greater proportion of positive headlines compared to neutral and negative ones. The distribution is as follows: 2134 positive headlines (61%), 789 neutral headlines (23%), 577 negative headlines (16%). This imbalance serves as a key consideration in evaluating the model families.
The study employs three different kinds of models that are fitted on the training data: classification, regression, and transformers. While classification and regression models predict values for both training and testing data, the transformer model is only used to predict for the testing data. Moreover, a stratified splitting technique is employed for the classifiers and regressors, while the transformer uses a standard split on the same data partition, as its fine-tuning process inherently handles imbalance during pre-training.

To prepare the data for the first part of the research, since we focused on traditional machine learning models, classification and ordinal regression alike, the sentiment labels were converted from categorical to numerical values. Specifically, the positive sentiment label was encoded as +1, neutral sentiment label as 0, and negative sentiment label as -1. This was done to ensure consistency across classification and regression tasks, as well as to make regression error values more meaningful and interpretable for direct comparison between model families.

In order to convert the headlines into numerical data for the machine learning models, the pre-trained BERT model (BERT-base-uncased) was used, along with its tokenizer (5). First, each headline was tokenized using the tokenizer, then passed through the pre-trained model to be converted into vector embeddings for every token in the headline. These context-based embeddings, extracted from the [CLS] token, were stored as numerical vectors and acted as the input features for the traditional machine learning models. Next, the dataset containing embeddings as input features (X_embedded) and encoded labels as target variables (Y_mapped) undergoes a stratified split into training and testing subsets such that 66% of the data is used for training and 33% is used for testing. Stratification is done to preserve the class imbalance observed in the dataset to enable fair evaluation of the models. A random state of 42 is set to ensure that the results can be reproduced, meaning the same split will occur every time the code is executed. (9) Hyperparameters for these models were optimized via GridSearchCV with 5-fold stratified cross-validation on training data only (9). In a 5-fold CV, the training set is divided into 5 equal parts, such that the model trains on 4 folds (80%) and tests on 1 fold (20%), repeating 5 times so each fold serves as validation once. GridSearchCV exhaustively evaluates all combinations of parameters from predefined grids to identify optimal settings that maximize macro F1 score (for classification) and minimize MSE (for regression). Final models, with optimal hyperparameters, are trained on the entire training set and evaluated on the test set.

The first part of the study employs six traditional machine learning models, three classification and three regression ones to predict headline sentiment on both the training and testing datasets. The classification models include Random Forest Classifier, Support Vector Machine, KNeighbors Classifier (9). The regression models include Linear Regression, Decision Tree Regressor, KNeighbors Regressor (9). The sentiment values are treated as discrete numerical values for classification models and continuous numerical values for regression models. The following provides a brief description of the models, their relevance to the problem statement, along with their performance metrics.

Classification Models:

- **Random Forest Classifier**
  A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. The parameter n_estimators refers to the number of trees in the forest whereas max_depth is the maximum depth of the tree. Additionally, the parameters min_samples_leaf meaning the minimum number of samples required to be at a leaf node and min_samples_split meaning the minimum number of samples required to split an internal node are used as well. In this paper, this model is tuned using n_estimators, max_depth, min_samples_leaf, and

min_samples_split to predict the sentiment of a headline in order to reduce overfitting that is often caused by only using a single decision tree. (9)

- **Support Vector Machine Classifier**
  A support vector machine classifier is a supervised learning method that separates data into classes by drawing the best possible boundary between them. It is effective for high dimensional feature spaces and was chosen due to the dense vector embeddings. The parameter C refers to the trade-off between correct classification and margin size, while gamma refers to how far the influence of a single training example reaches. Additional parameters kernel defines decision boundary type and class_weight upweights the minority negative class. In this paper, multi-class classification is used through the OvR (one-vs-rest) strategy (9).
- **KNeighbors Classifier**
  A k-nearest neighbors classifier is a non-parametric supervised learning method where the output is a class membership. The classifier assigns each headline to a class based on the most common among its k-nearest neighbors in the data (k is a positive integer, typically small) (9). The value of k is selected to balance sensitivity to local patterns with generalization across dataset. The parameter weights determines if closer neighbors get more influence while metric defines distance measurement. In this study, the classifier was tuned using various n_neighbors, weights, and metric values.

The performance of the hyperparameter-tuned classification model is evaluated using an accuracy score and macro-averaged F1 score, calculated for both training and testing datasets separately to assess generalization. The results are recorded in Table 1 and 2. The accuracy score of a classification model (9) measures the proportion of correctly predicted labels to the total number of predicted labels. The macro-averaged F1 score of a model is just a simple average of the class-wise F1 scores obtained (9).

Macro F1 score = $(1/n)\sum_{i=1}^{n} F1\ Score_i$

where n is the number of classes,

F1 Score$_i$ represents the individual scores of each class i.

As the model becomes better at dealing with the class imbalance, macro F1 increases linearly with improvements in per-class F1 scores, since it is a simple average over all classes. In contrast, however, accuracy can increase even if performance on majority classes remains poor, since it is dominated by the majority class. This makes it a less suitable metric for this task.

Regression Models:

- **Linear Regression**
  A statistical linear model that estimates the relationship between a scalar response (dependent variable) and one or more explanatory variables (regressor or independent variable). The model fits a straight line to the data by using coefficients or weights w = (w1, …, wp) that minimize the residual sum of squares between the targets observed in the dataset, and the targets predicted by the linear model. This straight line determines the best linear approximation of the relationship between the dependent and one or more independent variables (9) (10). In this paper, we use this model, without hyperparameters, to predict the headline sentiment as a continuous numerical value which was not rounded to preserve the continuous nature of the regression task.

- **Decision Tree Regressor**
  A statistical non-linear model that predicts numerical values using a tree-like structure. It operates by recursively dividing the data based on features (independent variables) that best reduce prediction error. Each internal node represents a if-else decision based on a feature, and each leaf node represents a predicted numerical value. In order to make a prediction, the decision tree traverses from the root node to the leaf node based on feature values. The final prediction is the average of the actual target values in that leaf node. The parameter max_depth signifies the maximum depth of the tree (11) and max_features refers to the number of features to consider when looking for the best split. Additionally, the parameters min_samples_leaf meaning the minimum number of samples required to be at a leaf node and min_samples_split meaning the minimum number of samples required to split an internal node are used as well. In this paper, this model is tuned using max_depth, min_samples_leaf, min_samples_split, and max_features to predict the headline sentiment as a numerical value. (9)

- **KNeighbors Regressor**
  A k-nearest neighbors regressor is a non-parametric supervised learning method where the target is predicted by local interpolation of the targets associated with the nearest neighbors in the training set. The regressor assigns a value to each headline based on the average of the sentiment scores among its k nearest neighbors in the data (k is a positive integer, typically small) (9). The value of k is selected to balance sensitivity to local patterns with generalization across dataset. The parameter weights determines if closer neighbors get more influence while metric defines distance measurement and p sets distance power. In this study, the classifier was tuned using various n_neighbors, weights, metric, and p values.

The performance of regression is assessed using mean squared error (9) and mean absolute errors (9) as evaluation metrics for both training and testing data separately. The results are recorded in Table 3 and 4. The mean squared error (MSE) and the mean absolute error (MAE) measure the amount of error in statistical models. The MSE assesses the average squared difference between the observed and predicted values whereas the MAE assesses the average absolute difference between the observed and predicted values.

$$\text{MSE} = (1/N)\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

$$\text{MAE} = (1/N)\sum_{i=1}^{N}|y_i - \hat{y}_i|$$

where N is the number of data points,

y_i represents actual values,

$\hat{y}_i$ represents predicted values.

As model error increases, the magnitude of increase in MSE is quadratically proportional to the magnitude of error. However, the magnitude of increase in MAE is directly (linearly) proportional to the magnitude of error.

The second part of the study employs a text-to-text transformer model, namely T5-small (7) to predict headline sentiment on both the training and testing datasets. To prepare the input dataset for the model, the columns containing headline and sentiment were flattened so that each headline and the corresponding sentiment label were represented as a single string value. Next, the input features (X) and target variables (Y) are split into training and testing subsets such that 66% of the data is used for training and 33% is used for testing. A random state of 42 is set to ensure that the results can be reproduced, meaning the same split will occur every time the code is executed.

A zero-shot evaluation was performed on the pre-trained T5-small model without any fine-tuning involved. Each headline, preceded by "sentiment analysis:" as the prompt, was tokenized using a T5 tokenizer. The predictions of the model were then decoded as text labels and sorted into one of three categories: positive, neutral, or negative. If an unexpected output was predicted, then the label defaults back to "neutral". The performance of T5-small's zero shot prediction on the test set is evaluated using accuracy score (12). This acted as a baseline assessment of the model's prediction ability without fine-tuning according to the task.

Next, the T5-small model is fine-tuned on the training dataset to improve its performance in sentiment prediction. Similar to zero-shot prediction, each headline, preceded by "sentiment analysis:" as the prompt, was used as input features while corresponding sentiment labels were used as target variables. Both inputs and targets were tokenized using a T5 tokenizer, but padding was applied to ensure uniform sequence lengths. An attention mask is used to indicate which tokens should be ignored (mainly for padded tokens). The tokenized version of target variables (sentiment labels) is assigned to the labels key in the inputs dictionary to serve as inputs to the decoder during training.

The T5-small model is then fine-tuned for different number of epochs (2, 5, 10) with a batch size of 8 and AdamW optimizer learning rate of 5×10^{-5} in order to update model parameters. After being set to training mode, each batch of tokenized inputs and corresponding labels were moved to the same place as the model, and the loss was computed automatically by the model. Backpropagation was used to calculate gradients and weights were updated after each batch. The training loss is also recorded for each iteration and visualized in a graph (Figure 1, 2, 3, 4).

In addition to fine-tuning for a fixed number of epochs, this study implemented a resource efficient strategy for training the model for up to 10 epochs but using an early stopping mechanism. To carry out this strategy, two variables are defined to track the total training loss after each iteration. After each epoch, the cumulative training loss sum is compared to the cumulative training loss sum for the previous epoch by subtraction. If the difference falls below a certain threshold value (arbitrarily chosen as 5% for this study), convergence condition is met and training is subsequently stopped. This approach allows us to conserve computational resources while ensuring that the model is trained properly. The training loss for each batch is recorded and visualized in a graph (Figure 1, 2, 3, 4).

After fine-tuning, the T5-small model was used to predict sentiment values on the test dataset. The model was set to evaluation mode and the predictions for each input were decoded as text labels and sorted into one of three categories: positive, neutral, or negative. Unexpected outputs were, once again, defaulted to the "neutral" label. This produced predictions for all input features (headlines) in the test data, which were then compared against the actual target variables (sentiments) to evaluate model performance. The performance of the fine-tuned T5-small transformer model is assessed using the following metrics:

- Accuracy measures the percentage of correct predictions for the test data. It is calculated as the ratio of true positives and true negatives to the sum of the total number of predictions. It treats false positives and false negatives equally. (13)
- Precision indicates the ratio of correct positive predictions. It is computed as true positives divided by the sum of true positives and false positives. It is concerned with the reliability of positive predictions. (13)
- Recall is the ability of a model to correctly identify the true positive. It is computed as true positives divided by the sum of true positives and false negatives. High recall means that the observed true positives are relatively higher than observed false negatives and low recall means the opposite. (14)
- F1 score: F1 is an overall measure of a model's accuracy that combines precision and recall, by taking the harmonic mean or the weighted average of precision and recall. (14)

The values of the performance metrics are recorded in Table 3 for different numbers of epochs. A classification report is generated, featuring the overall accuracy as well as precision, recall, and F1 score for each class.

This methodology provides a standardized framework for evaluating model performance across classification, regression, and transformer models. The next section presents the performance results of the evaluation metrics for all three types of machine learning models.

RESULTS

Classification Models

The following table (Table 1) presents the best cross-validated macro F1 score, macro-averaged F1 scores and the accuracy scores computed on the training and testing datasets for the three classification models evaluated in this study. The best values per column are highlighted in bold.

Model	Best CV macro F1	Train macro F1	Test macro F1	Test weighted F1	Training Accuracy	Testing Accuracy
Random Forest Classifier	**0.7269**	0.9848	**0.7763**	**0.82**	0.9863	**0.8312**
Support	0.6786	0.8966	0.7075	0.77	0.9155	0.7792

Vector Machine						
KNeighbors Classifier	0.7238	**0.9904**	0.7539	0.80	**0.9915**	0.8104

Table 1

The table below (Table 2) represents the classification reports of the traditional classifiers, alongside the hyperparameter combination that maximizes their macro-averaged F1 score.

Model	Best Hyperparameters	Class	Precision	Recall	F1 score
Random Forest Classifier	'class_weight': 'balanced', 'max_depth': 15, 'min_samples_leaf': 5, 'min_samples_split': 10, 'n_estimators': 200	negative	0.94	0.53	0.68
		neutral	0.91	0.67	0.77
		positive	0.80	0.97	0.88
Support Vector Machine	'C': 10, 'class_weight': None, 'gamma': 'scale', 'kernel': 'rbf'	negative	0.70	0.49	0.58
		neutral	0.83	0.61	0.70
		positive	0.78	0.92	0.84
KNeighbors Classifier	'metric': 'euclidean', 'n_neighbors': 11, 'weights': 'distance'	negative	0.78	0.55	0.65
		neutral	0.84	0.68	0.75
		positive	0.81	0.93	0.86

Table 2

Regression Models

The following table (Table 3) presents the best cross-validated mean squared error, mean squared errors (MSE) and mean absolute errors (MAE) computed on the training and testing datasets for the three regression models evaluated in this study. The best values per column are highlighted in bold.

Model	Best CV MSE	Training MSE	Testing MSE	Training MAE	Testing MAE
Linear Regression	0.7739	0.2366	0.5832	0.3672	0.5580
Decision Tree	0.5189	0.4567	0.5089	0.5540	0.5872

Regressor					
KNeighbors Regressor	**0.3661**	**0.0125**	**0.3247**	**0.0154**	**0.3370**

Table 3

The table below (Table 4) represents the classification reports of the ordinal regressors, alongside the hyperparameter combination that minimizes their mean squared error.

Model	Best Hyperparameters	Class	Precision	Recall	F1 score
Linear Regression	(no hyperparameters to tune)	negative	0.61	0.36	0.45
		neutral	0.37	0.66	0.47
		positive	0.80	0.65	0.72
Decision Tree Regressor	'max_depth': 5, 'max_features': 'log2', 'min_samples_leaf': 5, 'min_samples_split': 5	negative	0.89	0.09	0.16
		neutral	0.30	0.53	0.39
		positive	0.71	0.68	0.69
KNeighbors Regressor	'metric': 'euclidean', 'n_neighbors': 15, 'p': 1, 'weights': 'distance'	negative	0.93	0.50	0.65
		neutral	0.49	0.78	0.61
		positive	0.84	0.77	0.80

Table 4

Transformer Model (T5-small)

Zero-shot prediction accuracy = 0.2165 (defaults to neutral)

The following table (Table 5) shows the overall accuracy as well as the precision, recall, and F1 score for each category for T5-small model when predicting the headline sentiment. The metrics were directly obtained from the classification report. The result highlighted in bold represents the most optimal configuration for the model.

Epochs	Class	Accuracy	Precision	Recall	F1 score

2	negative	0.61	0.44	0.02	0.04
	neutral		0.58	0.16	0.26
	positive		0.61	0.96	0.75
5	negative	**0.74**	0.76	0.45	0.57
	neutral		0.73	0.51	0.60
	positive		0.74	0.91	0.81
10	negative	0.71	0.59	0.57	0.58
	neutral		0.70	0.49	0.58
	positive		0.75	0.84	0.79
epochs when stopped at 5%	negative	0.65	0.93	0.07	0.12
	neutral		0.80	0.24	0.37
	positive		0.64	0.98	0.77

Table 5

The following graphs visualize the training loss of the T5-small model for different numbers of epochs.
For 2 epochs,

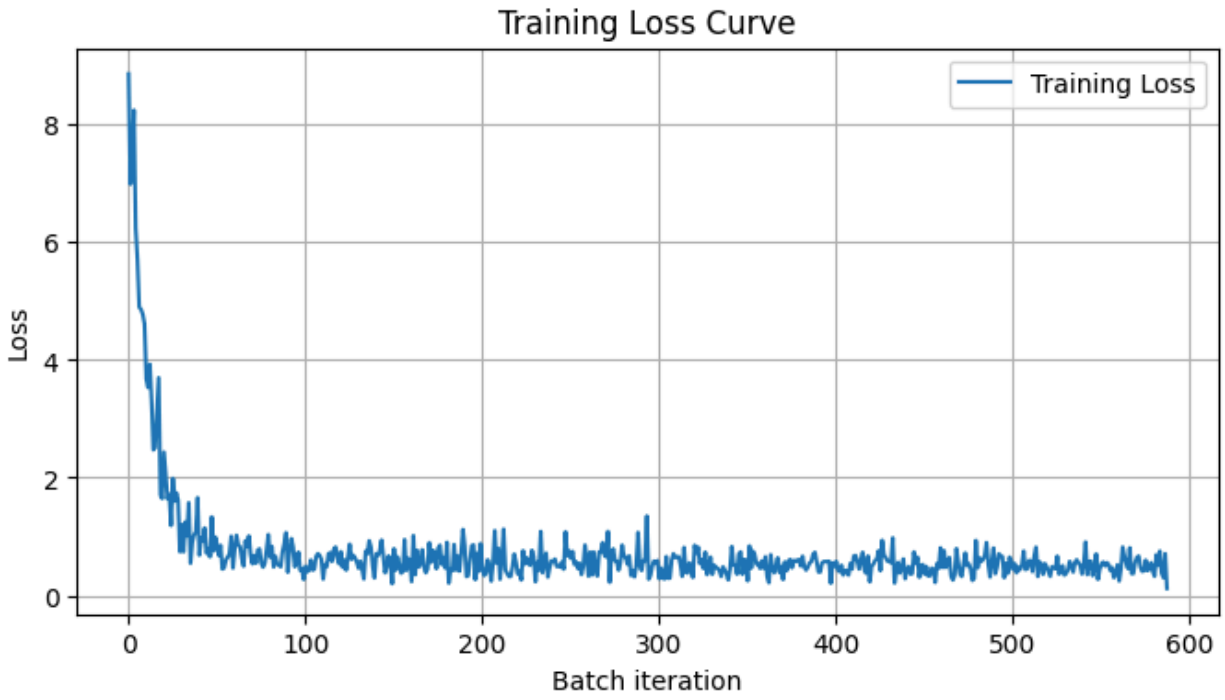

Figure 1
For 5 epochs,

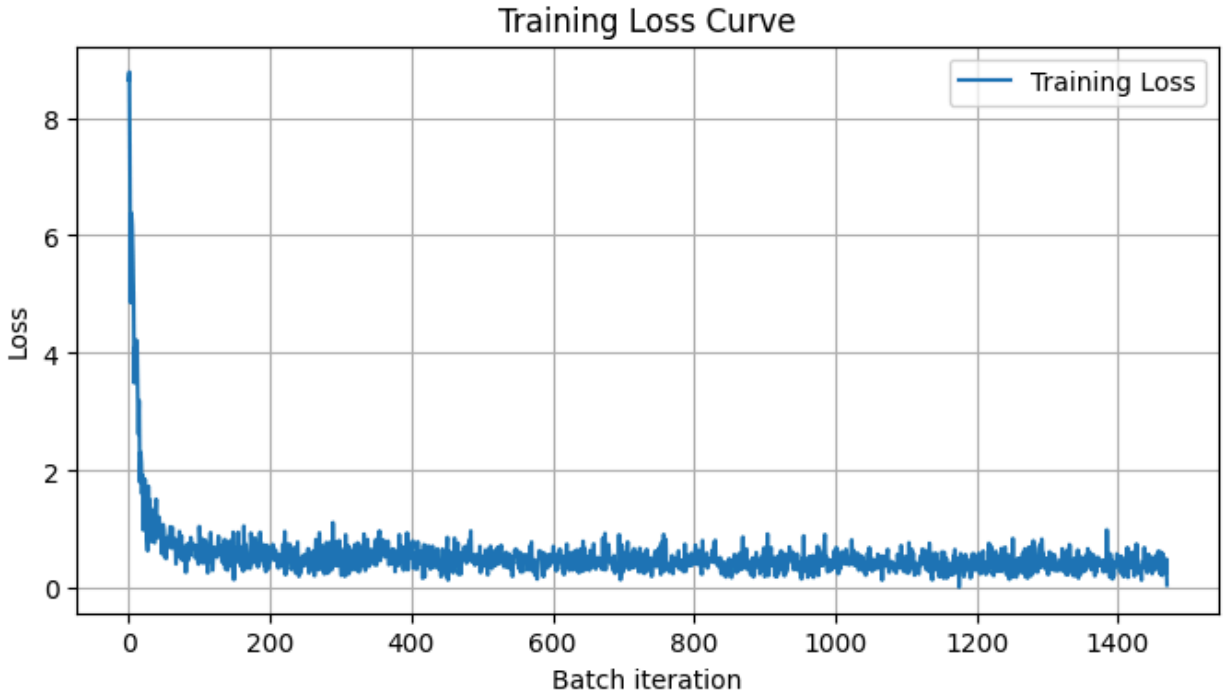

Figure 2
For 10 epochs,

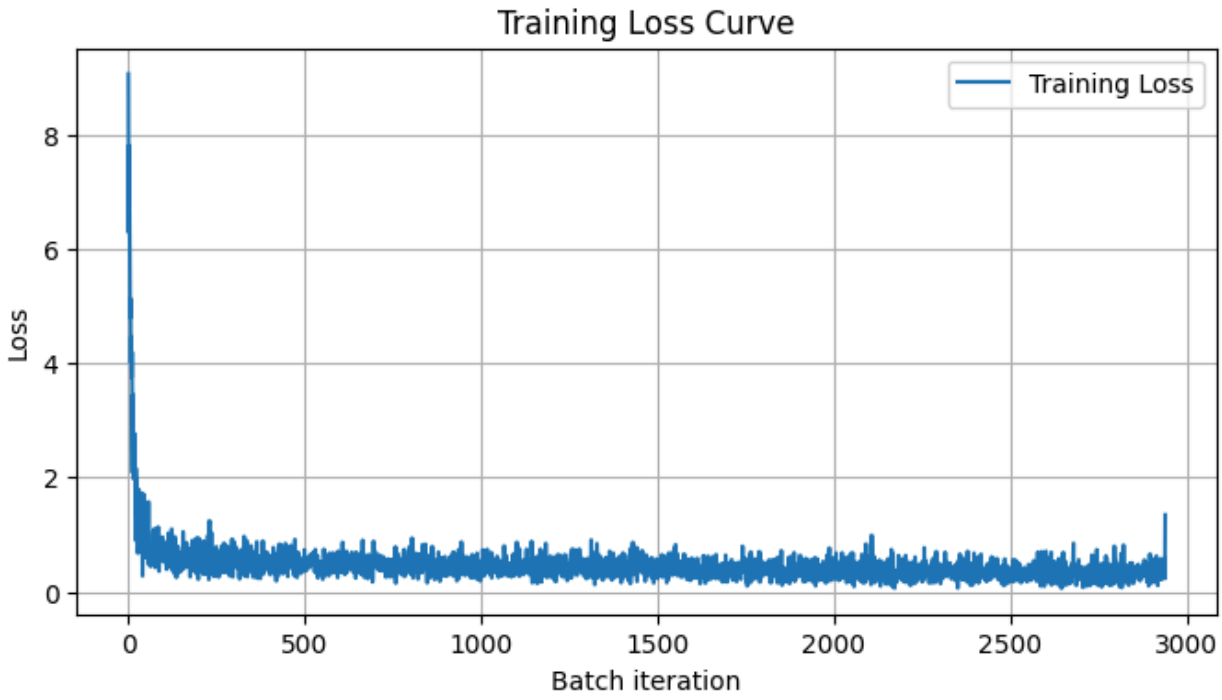

Figure 3

For epochs until breakage <0.05,

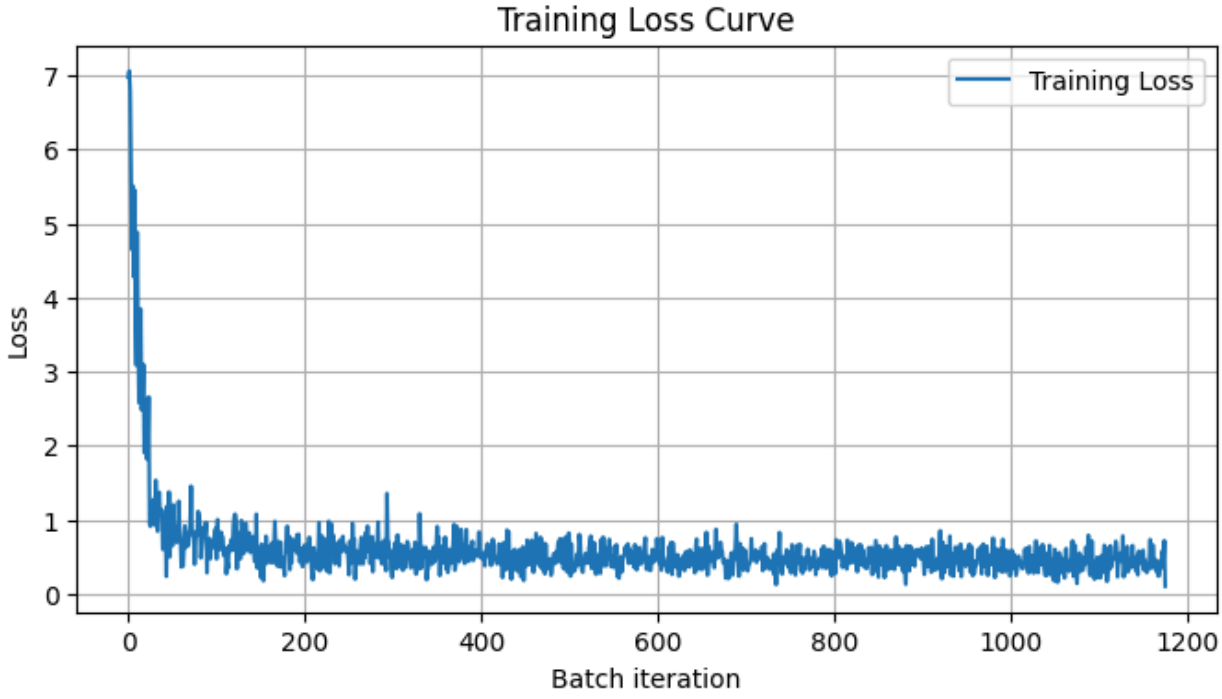

Figure 4

DISCUSSION

This study demonstrates the performance of machine learning models in predicting the sentiment labels of a New York Times headline. The models demonstrated varying degrees of agreement with TextBlob labels, which served as the baseline for lexicon-based sentiment predictions. Three classification models (Random Forest Classifier, Support Vector Machine, KNeighbors Classifier), three regression models (Linear Regression, Decision Tree Regressor, KNeighbors Regressor), and one transformer model (T5-small) were developed and compared within their types using performance metrics on testing sets.

Among the traditional classification models shown in Table 1, the Random Forest Classifier demonstrated the best generalization abilities, with the highest test macro F1. However, there is a slight overfitting issue as seen by the drop in performance from training to testing, in both macro F1 and accuracy scores. Despite the KNeighbors Classifier achieving the highest training values, the individual F1 scores in Table 2 indicate its tendency to overfit to the positive majority class. Support Vector Machine has the lowest metric values but tends to generalize better (lesser difference between training and testing data). All classifiers maintained strong test performance, with the 5-fold CV scores serving as reliable predictors of the final test results, justification for the stratified cross-validation approach for this imbalanced dataset. Random Forest's high CV score directly translates to best performance, suggesting optimal hyperparameter selection by GridSearchCV.

For the regression models in Table 3, the testing error for all three models is quite high, making them unsuitable for sentiment prediction tasks. KNeighbors Regressor exhibits extreme overfitting, with a dramatic jump (25-fold) between training and testing MSE/MAE, while Decision Tree Regressor's minimal CV-to-test gap suggests optimal hyperparameter tuning by GridSearchCV. However, the individual F1 scores in Table 4 show the strongest per-class performance for KNeighbors Regressor, particularly for minority classes. This demonstrates that KNeighbors Regressor substantially outperforms Decision Tree Regressor, even while memorizing training data. Although unsuitable by regression error metrics, KNeighbors Regressor achieves higher per-class F1 after discretization. Linear Regression demonstrates stable generalization yet high errors, exhibiting average performance.

The zero-shot accuracy of the T5-small model is 0.2165, which closely aligns with the proportion of neutral sentiment labels in the dataset (≈23%), suggesting that the model tends to default to predicting the neutral class rather than providing meaningful sentiment classification.

The results of the T5-small model in Table 5 indicate that the number of fine-tuned epochs has a significant impact on the model's performance. At 2 epochs, the model seems to be undertrained, with very low recall for negative and neutral classes but high for the positive class. This demonstrates a bias for positive sentiment, which could be a result of the previously mentioned imbalance in the dataset. The training loss curve (Figure 1) shows a significant decrease initially, but the increased fluctuations at a relatively high loss value and premature convergence indicate strong underfitting that makes it unsuitable for sentiment prediction tasks. However, at 5 epochs, the model shows the best overall accuracy and a much more balanced precision and recall across the three classes. The training loss curve (Figure 2)

shows rapid convergence and stabilization at a relatively low point, indicating sufficient learning that leads to the best observed performance. It is important to note that, even here, the precision, recall, and F1 score are highest for positive labels due to the bias intrinsic to the dataset itself.

At 10 epochs, the overall accuracy, neutral recall, and neutral precision drop slightly while negative recall and negative precision worsen. This indicates that the model has started to overspecialize, instead of gaining meaningful generalizations. The training loss curve (Figure 3) shows a great drop and quick stabilization, but the absence of further loss reduction indicates that the performance has plateaued, which, along with the declining evaluation metrics, shows the presence of overfitting. In the special case where the model is stopped once it drops below 5% training loss, the number of epochs seems to be a little less than 5, as seen in Figure 4. Here, the model is stopped too early, and an imbalance arises among the classes, indicating underfitting. The training loss curve (Figure 4) shows a premature termination before stabilization (since the variance is still high), indicating insufficient learning and possible bias. In order to improve performance, the threshold value could be reduced to ensure the model has learned sufficiently before stopping. Inferring from the results above, training the model with 5 epochs seems to be the optimal choice, as there is a good generalization and balance exhibited across classes.

Comparing across all three model families, the Random Forest Classifier outperforms the fine-tuned T5-small model and all the ordinal regression models with a test macro F1 of 0.7763. While classification models generally performed better than regression ones, KNeighbors Regressor surprisingly excelled on minority classes despite overfitting. Although T5-small with 5 epochs is fairly competitive, it tends to favor the majority class and exhibit limited capability to generalize across other classes, as seen in the lower recall and F1 scores for neutral and negative sentiment classes. In contrast, fine-tuning the classification and regression models greatly improved performance, achieving a better balance across all sentiment classes. This demonstrates that BERT embeddings enable a strong baseline across architectures, although a 16% minority class remains challenging for all model families.

Despite these findings, the results are subject to certain limitations. This analysis is limited to New York Times headlines from one week in July 2024, representing a narrow time window that may not capture variation in sentiment patterns across different time periods and news cycles. The study relies on lexicon-based sentiment predictions by TextBlob as the baseline, which might not be accurate owing to the limited context in headlines. Moreover, the dataset imbalance introduces a bias towards the majority class in predictions, potentially limiting generalization and affecting model performance.

CONCLUSION

This study evaluates the performance of traditional machine learning models and transformer-based architecture for sentiment analysis of imbalanced New York Times headlines data. BERT embeddings were used as input features to assess hyperparameter-tuned classification and ordinal regression models alongside a T5-small text-to-text transformer model in both zero-shot and fine-tuned settings.

According to the performance metrics listed in results, Random Forest Classifier performed best among all model families, establishing traditional machine learning as a surprisingly good baseline that outperformed fine-tuned T5-small and all regression models. Notably, the KNeighbors Regressor excelled on minority classes despite overfitting, while transformers showed competitive potential at optimal epochs but may encounter underfitting at premature early stopping, sacrificing minority class performance. Stratified cross-validation with GridSearchCV optimization proved essential for handling class imbalance, validating the -1/0/+1 encoding strategy across model families. The zero-shot transformer model struggled to provide meaningful predictions, majorly defaulting to neutral class, highlighting the challenge faced when directly applying models to problems without fine-tuning. Fine-tuning T5 showed significant improvements in performance, with 5 epochs identified as the most optimal balance between adequate learning and overfitting. At this configuration, the model achieved the highest accuracy and a relatively balanced precision, recall, and F1 score across three classes, despite intrinsic bias due to dataset imbalance. Training with less or more epochs led to underfitting or overfitting, respectively, emphasizing the importance of hyperparameter selection.

Overall, these findings challenge the assumption that transformer-based fine-tuning outperforms traditional classification and regression models for short text sentiment tasks. Future work in this field can aim to explore other machine learning models, potentially more complex ones to develop a more comprehensive and accurate prediction. Hybrid approaches combining BERT-embeddings with transformer architecture or advanced resampling techniques can be explored to improve minority class performance. Fine-tuning hyperparameters in transformer models can further be refined by decreasing the training loss threshold value. Additionally, using larger architectures such as T5-medium or T5-large can greatly improve performance. Expanding the dataset to cover a longer period of time or incorporating more context (articles instead of headlines) can improve the model's ability to predict the sentiment accurately. Such improvements will be a valuable boost to the ongoing research within this field.

REFERENCES

1. Wankhade, Mayur, Annavarapu Chandra Sekhara Rao, and Chaitanya Kulkarni. "A survey on sentiment analysis methods, applications, and challenges." Artificial Intelligence Review 55.7 (2022): 5731-5780.
2. Liu, Bing. Sentiment analysis and opinion mining. Springer Nature, 2022.
3. Dor, Daniel. "On newspaper headlines as relevance optimizers." Journal of pragmatics 35.5 (2003): 695-721.
4. Singh, Anurag, and Goonjan Jain. "Sentiment analysis of news headlines using simple transformers." 2021 Asian Conference on Innovation in Technology (ASIANCON). IEEE, 2021.
5. Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers). 2019.
6. Loria, Steven. TextBlob Documentation. Version 0.15, 2018.

7. Raffel, Colin, et al. "Exploring the limits of transfer learning with a unified text-to-text transformer." Journal of machine learning research 21.140 (2020): 1-67.
8. clovisdalmolinvieira. News Sentiment Analysis. Kaggle, 2024, https://www.kaggle.com/datasets/clovisdalmolinvieira/news-sentiment-analysis
9. Fabian, P., 2011. Scikit-learn: Machine learning in Python. Journal of machine learning research 12, p.2825.
10. McCarthy, R.V., McCarthy, M.M. and Ceccucci, W., 2022. Predictive models using regression. In Applying Predictive Analytics: Finding Value in Data (pp. 87-121). Cham: Springer International Publishing.
11. Czajkowski, M. and Kretowski, M., 2016. The role of decision tree representation in regression problems–An evolutionary perspective. Applied soft computing, 48, pp.458-475.
12. Xian, Yongqin, Bernt Schiele, and Zeynep Akata. "Zero-shot learning-the good, the bad and the ugly." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.
13. Naidu, Gireen, Tranos Zuva, and Elias Mmbongeni Sibanda. "A review of evaluation metrics in machine learning algorithms." Computer science on-line conference. Cham: Springer International Publishing, 2023.
14. Obi, Jude Chukwura. "A comparative study of several classification metrics and their performances on data." World Journal of Advanced Engineering Technology and Sciences 8.1 (2023): 308-314.