www.editada.org

International Journal of Combinatorial Optimization Problems and Informatics, 15(5), Dec 2024, 6-16. ISSN: 2007-1558. https://doi.org/10.61467/2007.1558.2024.v15i5.554

Machine learning techniques for sentiment analysis

Jessica Olivares Lopez, Abraham Sánchez López, Rogelio González Velázquez, María del Carmen Santiago Díaz, Ana Claudia Zenteno Vázquez

Benemérita Universidad Autónoma de Puebla. de Ciencias de la Computación, Av. 14 Sur y San Claudio, Col. San Manuel, CP 72570, Puebla, Pue. México
olivares.lopez.jessica3p@gmail.com, abraham.sanchez@correo.buap.mx, rogelio.gonzalez@correo.buap.mx, marycarmen.santiago@correo.buap.mx, ana.zenteno@correo.buap.mx.

Abstract. Sentiment analysis stands out as one of the most dynamic and pivotal areas in the field of natural language processing. In this work, a range of machine learning strategies has been proposed, applied, and benchmarked for sentiment analysis, with a specific focus on supervised machine learning techniques. Various algorithms have been considered and applied to texts extracted from Twitter. Furthermore, the results are compared with works that applied unsupervised machine learning techniques to the same dataset.

Keywords: Sentiment analysis, Machine Learning, X, Document classification.

Article Info
Received May 1, 2024.
Accepted Nov 20, 2024.

1 Introduction

Sentiment analysis is a type of document classification that leverages artificial intelligence techniques, particularly machine learning, to assign a class or label based on the structure and content of a document, sentence, or text in general. This process involves rigorous analysis and text processing.

Text classification for sentiment analysis can have different approaches. One approach is based on 'sentiment polarity', where a sentence or document is assigned a polarity that can take various values: negative, positive, or neutral. Another approach is sentiment analysis through emotion classification, where labels are assigned based on the emotions identified in the sentence or text.

Nowadays, sentiment analysis has become a powerful tool, owing to its diverse applications in various industries. It is utilized in different approaches such as business intelligence, marketing, decision-making, and support assessment, among others, making it applicable to a wide range of sectors. This evaluation method is now considered crucial for any company, with the primary goal of extracting the maximum value from data. The way data is processed has become a differentiating factor in the industry. Data collected in daily processes can yield numerous deductions, conclusions, and enriching insights that validate and even influence crucial decision-making.

This work presents an evaluation of a set of supervised algorithms for sentiment analysis, considering various strategies based on the assessment of different types of algorithms, including those based on distance, ensemble, and more advanced algorithms such as perceptron and Ridge Classifier. These diverse algorithms are applied

to the same dataset, aiming to maintain the essence of sentiment analysis and to obtain an assessment of the varied results stemming from the utilization of both unsupervised and supervised techniques.

2 Related Work

In the literature, numerous works focus on supervised processes. However, the following works were explored in more detail because their results are based on the performance of models generated with the algorithms selected for exploration.

Sentiment Analysis Using Support Vector Machines, Neural Networks, and Random Forests

The approach in (Wang, 2023) offer distinct methodologies for sentiment classification facilitating their effective application in real-world scenarios, each with unique strengths and applications. As sentiment analysis evolves, exciting developments are anticipated to improve the understanding of sentiments in textual data, enabling better decision-making and engagement across various domains. In this work, the author employs SVM and Random Forest algorithms for sentiment analysis but also leverages more robust deep learning algorithms such as Neural Networks.

Sentiment Analysis using Support Vector Machine and Random Forest

In comparison to previous work (Wang, 2023), the study by Khan et al. (2024) applies the theory presented to sentiment analysis based on sentiment polarity. They utilized a manually compiled dataset and incorporated the step of crawling technology articles from reputable and trustworthy news websites spanning over a decade. The research paper provides an overview of sentiment analysis, discussing preprocessing techniques, feature extraction methods, and six machine learning techniques: Naive Bayes, Support Vector Machines, Decision Trees, Random Forests, Neural Networks, and Unsupervised Learning approaches. It also covers model training, evaluation metrics, and the importance of comparative analysis, helping researchers apply sentiment analysis across various domains.

Additionally, the paper highlights a significant trend of negative sentiments within the analyzed comment dataset, suggesting further exploration of the factors contributing to this negativity to gain a comprehensive understanding of the sentiment landscape and uncover relevant patterns and themes.

Comparison of Parameters of Sentimental Analysis Using Different Classifiers

This study presents the results of various detection models generated from 10 different algorithms using the sentiment140 dataset. The data cleaning process is briefly outlined, followed by a comparison using metrics such as Accuracy, Precision, Recall, and F1 score. An interesting finding is that one of the top-performing algorithms based on Accuracy is the Ridge classifier, which is also explored in this work.

Strategies for sentiment analysis of texts extracted from Twitter using deep learning techniques

In this work (Olivares, 2022), a series of deep learning strategies is proposed for sentiment analysis problems, focusing on unsupervised techniques. The data processing aims to gain insights into artificial intelligence topics such as Machine Learning and Deep Learning. Some of the explored strategies include bag-of-words, word embeddings, and other techniques for text preprocessing in Natural Language Processing (NLP).

Finally, the results highlight intriguing insights into the public opinion of artificial intelligence in X. This is achieved through the interpretation of elements such as word clouds, noun analysis, intelligent summaries, and emotion analysis.

Bag-of-words is a popular and simple technique for extracting features in texts for sentiment analysis. It is responsible for transforming each text input (document, sentence) into a vector and each word in this input goes through a scoring process, where each score is assigned in the location equivalent to the representation. This evaluation or scoring process can be applied by different methods: Binary, Counting, TF-IDF and Frequency.

Word embeddings is a representation of text that starts from the idea words with similar or equivalent meanings have similar vector representations. They are represented by vectors in a previously defined vector space.

A vector is designated for each word, the vector values go through a training process, so that this training process is like that applied to a neural network. By passing through this training word embeddings can represent more information in a smaller number of dimensions.

One of the important objectives in the proposal is the retrieval of X data, so that the integration of the dataset meets the proposed characteristics, as it is composed of text (unstructured inputs) it is necessary to use preprocessing techniques before carrying out the sentiment analysis. Therefore, the quality of integration of this dataset with the structure and information required to carry out the required process is important. The consolidation of the dataset used in this work was carried out in three stages: acquisition or retrieval of the dataset, data cleaning and data labeling.

For the proposed cleaning procedure, two processes are proposed: basic cleaning and text cleaning for NLP. The initial cleanup lies in the exclusion of certain elements that initially made up the corpus and that can generate noise to the models. Extracting observations from Twitter naturally retrieves additional text elements such as: usernames, hashtags, mentions, URLs, and emoticons. These elements were eliminated to avoid introducing noise and reducing the performance of the models to be evaluated. On the other hand, there is the cleaning of the data in text format for NLP that proceeds in the following phases:

- Convert text to lowercase.
- Remove double spaces.
- Delete numbers. This is very important because it prevents inaccurate interpretation of numbers in the models.
- Delete StopWords. Eliminate short words that are used periodically and usually do not add value to the context, some examples are: prepositions, conjunctions, articles, among others.
- Stemming. This process transforms each word into its corresponding base form.

Labeling the data set is a highly relevant process, allowing supervised techniques to be used for text analysis. Generally, this process is done manually and under the supervision of experts. Once the dataset was collected and structured, a label was assigned; in this case, a classification based on sentiment polarity was chosen. This data labeling phase was performed in a semi-supervised manner, using the Azure Text Analysis resource.

During data exploration, it is about understanding the composition of the data and achieving through the analysis of the data, interpretations that add value to the task of sentiment analysis. NLP has a variety of techniques that help visualize and interpret data (text) efficiently. Therefore, some techniques were adopted, which due to the brevity of the document could not be explained in detail, some of them are mentioned.

In the first instance, two histogram graphs were made based on the occurrence of words, performing n-gram tests (2 and 3 grams). By noticing in the comparison of the two histograms, the implementation of bigrams helps the interpretation of the data from the context, better contextual coherence is obtained compared to the division into trigrams where less relevance of the subsets of words is observed. With the list of bigrams generated, it is possible to generate a semantic representation that favors the interpretation of the relationship in the different bigrams. Additionally, the visualization of bigrams favored the conclusion that the data set is assembled based on the expected content.

With the use of an Embeddings projector, TensorBoard (a tool that graphs vectors of word embeddings), for the interpretation of semantic relationships of the embeddings, it is possible to use a certain word as a vector representation and identify the closest words, based on the definition of Embedding words, similar or similar words based on the distance calculation, TensorBoard allows you to calculate the cosine or Euclidean distance.

An important aspect to highlight is the selection and balancing of data. As can be seen in the results for each sentiment category, there is a disproportion of observations, that is, the distribution of data is unbalanced, and this could affect the performance of the models by introducing information bias. To avoid this problem, the use of class balancing techniques is proposed. For this problem, it is desirable to equalize the distribution, i.e., to obtain the same number of observations for each of the sentiment polarities.

3 Models and Methods

In this proposal, in contrast to (Olivares et al., 2023), the decision was made to classify documents based on sentiment polarity using the same dataset. We implemented models from supervised algorithms, briefly described in the following sections. The aim is to gain insight into the various scopes provided by different techniques, encompassing both supervised and unsupervised approaches.

The algorithms chosen for this analysis have been strategically selected to cover diverse techniques, including ensemble learning through random forest, Ridge Classifier based on Ridge regression, and Support Vector Machine (SVM).

3.1 Random Forest

Random Forest is an algorithm composed of a set of trees, collectively known as a random forest. It acquires predictions from each individual tree and determines the class with the highest number of votes, employing an ensemble strategy. The algorithm generates multiple classifiers from smaller subsets of the input data, and their individual results are aggregated through a voting mechanism to produce the final output for the input dataset. Despite its straightforward nature, Random Forest is widely acknowledged as one of the most powerful machine learning algorithms available (Géron, 2017).

Random Forest algorithms essentially have three hyperparameters that must be set before the training process. These include the size of the nodes, the number of trees, and the number of features sampled. Depending on the problem to be solved, the random forest classifier can be used for regression or classification tasks.

In this work, the RandomForestClassifier from sklearn is employed as a classifier. Each sentiment polarity is considered as a class or label, which is then assigned to each observation. In this context, each observation corresponds to a Tweet from our dataset.

Unsupervised learning presents several significant challenges in research. Nonetheless, numerous algorithms have been developed to effectively learning data, yielding promising results. For instance, (Olivares et al., 2023) showcases some of these models applied to the same dataset used in this work, tweets with textual content associated with Artificial Intelligence.

The assessment and exploration of these models have revealed various components of sentiment analysis and information processing, providing valuable insights for decision-making based on the analysis. Strategies such as the utilization of KNN for data exploration, word embeddings, and others contribute to a more detailed understanding of the data composition.

On the other hand, the supervised learning evaluated in this work, employing Random Forest, Ridge Classifier, and SVM algorithms, yields more precise results. In other words, the metrics obtained are more relevant. Unlike the results in (Olivares et al., 2023), our approach achieves accuracy values exceeding 80%, and these results are obtained in shorter training times.

3.2 Ridge Classifier

Ridge regression is a linear model commonly used for regression, like ordinary least squares. However, it can also be adapted for use as a classifier with some additional modifications. In Ridge regression, the coefficients (w) are selected not only to predict well on the training data but with an additional constraint. The objective is to minimize the magnitude of the coefficients; in other words, all entries of (w) should be as close to 0 as possible.

In the context of classification, the Ridge Classifier aims to find the optimal hyperplane that separates different classes while considering the regularization term. This regularization term penalizes large coefficients, helping to control the model's complexity and reduce the risk of overfitting. The Ridge regression classifier initially converts the target values into {-1, 1} and treats the problem as a regression task (multi-output regression in the

multiclass case), optimizing the same objective as mentioned earlier. In this approach, the predicted class corresponds to the sign of the regressor's prediction. In the case of multiclass classification, such as in this sentiment analysis problem, text classification is treated as multi-output regression, where the predicted class corresponds to the output with the highest value.

3.3 Support Vector Machine

The Support Vector Machine (SVM) algorithm is a statistical classification approach based on the maximization of the margin between instances and the separation hyperplane. It is a non-probabilistic binary linear classifier that could linearly separate classes by a large margin. As a result, it has become one of the most powerful classifiers capable of handling infinite-dimensional feature vectors (Anrani & Lazaar, 2018). The basic idea behind SVM is to find the hyperplane that best separates different classes in the feature space.

4 Metrics

It is essential for each machine learning model to be able to implement quantitative measures that allow measuring the prediction quality being obtained. Metrics in machine learning are quantitative strategies that measure the performance of the models. There is a variety of metrics, with different specifications, considerations, parameters, and specific objectives, which determine which metric is appropriate to evaluate a certain algorithm. This section briefly showcases the metrics that have been used to evaluate the performance of the implemented algorithms.

4.1 Accuracy

Accuracy, a metric used to evaluate classification models, measures the proportion of correct predictions by dividing the number of correct predictions by the total number of predictions. This metric is a good option when working with datasets that have the same number of observations for each class.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

4. F1 Score

The F1 Score is the harmonic mean of precision and recall, measuring the accuracy of the test. The value ranges between 0 and 1. This metric helps balance precision and recall values, providing a more realistic idea of the model's behavior. Recall or sensitivity is the percentage of correct items that are selected. Recall equal to 1 means that all the positive examples were found. It represents how many times the model predicts a true positive correctly.

$$F1\ Score = \frac{Precision * Recall}{Precision + Recall} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

5 Implementation of strategies for sentiment analysis

There are diverse techniques set for sentiment analysis, encompassing tasks such as dataset processing, feature extraction, and feature representation, among other aspects within natural language processing. This study specifically emphasizes the classification of documents (tweets) based on sentiment polarity. The primary

approach involves creating classification models using supervised algorithms, as elaborated in the preceding section. To facilitate this, it is essential to have a labeled dataset for supervised training, which is further discussed in this section.

This section presents the general details of the training processes for each generated model, the performance benchmarks of each model, and, finally, an interpretation of how the model with the highest accuracy in predicting sentiment polarity obtains sentiment polarity for each observation from the sparse features compression of the training dataset.

5.1 Dataset

This section provides a brief overview of the structure and integration of the dataset used in classification models based on sentiment polarity. It includes a description of significant data preprocessing processes such as dataset creation, cleaning, and data balancing.

The dataset creation process involves three stages: data acquisition, data cleaning, and data labeling, which are detailed in [1]. From this process, 12,000 observations or tweets were obtained, distributed across sentiment categories as follows: neutral (6162), positive (2965), negative (2290), and mixed (583). Notably, observations were strategically retrieved to collect tweets with verbatim mentions of the terms:

- Artificial Intelligence
- Machine Learning
- Deep Learning

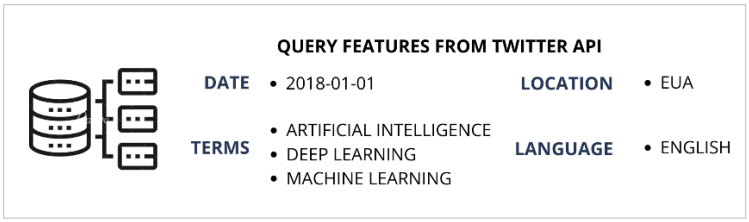

Figure 1. The specifics of the query employed to retrieve data through the Twitter API.

It's crucial to emphasize that data cleaning and preprocessing are integral steps in executing the algorithms subject to assessment and/or evaluation. These two steps have been meticulously performed and detailed at a technical level in (Olivares, 2022). For illustrative purposes, the process is depicted in the following image.

In the data balancing process, the goal was to balance the number of observations for each class. Due to the disparity in the number of observations, under sampling techniques were chosen, because generating synthetic data, particularly for the minority class, which is 10.57 times smaller than the class with the most observations, could introduce information bias over the majority and minority classes.

By applying random undersampling, we obtained 2,290 observations for each class. Additionally, the "mixed" class was removed because it represented less than 5% of the original dataset. The table below illustrates the final data set structure after processing applied.

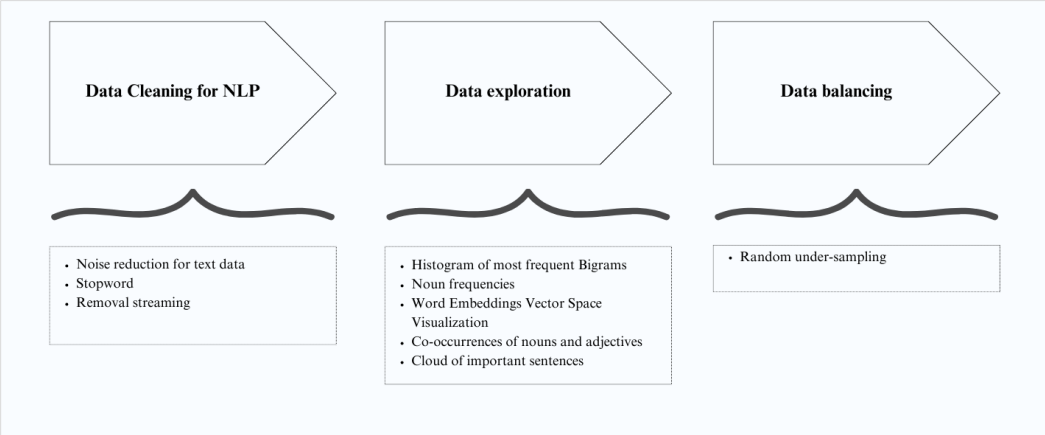

Figure 2. Data processing diagram applied to the dataset.

Table 1. Data set structure

Dataset	
Id	Twitter identification number
Text	Tweet content
CleanTweetS	Twitter content after the cleaning process
Sentiment	Sentiment polarity assigned to the content of the Twitter.
negative_score	Negative score of Twitter content.
positive_score	Positive score of Twitter content.
neutral_score	Neutral score of Twitter content.

5.2 Sparse features for sentiment analysis

One of the most astonishing challenges in NLP has been, over time, the development of techniques to find numerical representations that enable the evaluation and processing of text using algorithms based on mathematical models. Sparse features play a crucial role in Natural Language Processing (NLP), where data representations often involve high-dimensional and sparse feature spaces. In the context of this work, sparse features serve as a representation for model classifiers and data processing techniques.

The use of sparse features for processing the data in this work was undertaken with the objective of understanding the data composition from a contextual perspective. To achieve this, training vectors were developed using word embedding representation. Subsequently, the word embedding vector space was visualized using Projector Embeddings in TensorBoard (Embedding Projector, 2024). This technique made it easy to validate that, from the context of the initial problem, the observations in the dataset (Tweets) were faithfully formed around the proposed themes, proving to be revealing for Artificial Intelligence topics.

To facilitate understanding of this discovery, Figure (3) has been included, depicting the vector spaces of word embeddings in proximity to artificial and learning. This approach allows for the graphical visualization of embedding vectors, where each point represents the vector corresponding to a word's representation with its label. Words that are closest by cosine distance are considered contextually related.

Figure 3. (a) Vector space representation of word embeddings close to 'artificial'. (b) word embeddings close to 'learning'

In addition to using TF-IDF to assign weights to words based on their frequency in a document and their rarity across the entire dataset and training the models with these vectors (Wartler, 2017, July 24), the vectors generated by TF-IDF were used for manual inspection of classified documents. This approach aimed to gain insights into the data and develop a deeper understanding of how these classifiers make decisions by examining the words with the highest average feature effects.

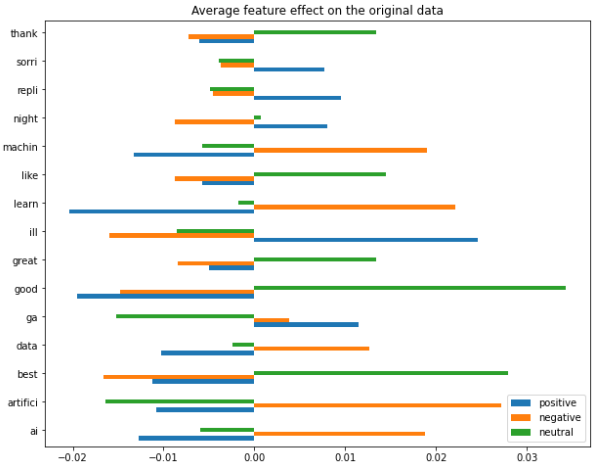

Figure 4. Average feature effect on the original data.

Analyzing the average feature effect on the original data provides intriguing insights that were not previously discovered in this labeled dataset. Terms related to Artificial Intelligence, such as artificial, learn, machine, ai, and data, when observed in the training set, are strongly positively associated with the negative class or polarity.

5.3 Implementation of algorithms for generating sentiment analysis models

The generation of classification models based on sentiment polarity in this work is divided into two processes: training and testing. These processes utilized scikit-learn (Scikit-learn, 2024), employing available classifiers according to the detailed algorithms.

The training and testing process of the three different classification models, it was necessary to partition the dataset into train and test sets. This division is a crucial step in any Machine Learning tasks. Working with distinct samples in both processes ensures that the test validates the robustness and performance of the model with unseen data from what was used in training. Specifically, the division corresponds to 70% for the training set and 30% for the test set.

Each algorithm was independently evaluated using specific parameters tailored to it, with the same dataset, training set, and test set, to simulate consistent conditions. The final performance results were measured using Accuracy and F1 Score metrics.

Table 2. Metrics of experimental results for Random Forest, Ridge Classifier, SVM classifier with. The best results are highlighted in bold.

Dataset	Accuracy	F1 Score
Random Forest	0.789	0.791
Ridge Classifier	**0.825**	**0.825**
SVM	0.817	0.817

In this case, the results of the work partly share the performance of the models as in (Khan et al., 2024), for SVM and Random Forest. The experimental findings indicate that the Support Vector Machine (SVM) algorithm outperforms the Random Forest algorithm in terms of fitting degree and generalization ability for sentiment analysis. SVM has superior performance in accuracy and F1-score, as well as its ability to handle complex features and relationships between words and spaces feelings. Random Forest, while also a strong performer. SVM excels in handling high-dimensional data and generalizing to new data.

Based on the results presented in Table (2), it is evident that the Ridge Classifier exhibited the highest performance, achieving a superior ranking across all three evaluated metrics compared to Random Forest and SVM. Notably, it demonstrated this high performance while requiring the shortest training time. Consequently, was decided to showcase the predictive capacity for different sentiment polarities positive, neutral, and negative by analyzing the confusion matrix illustrated in Figure (5).

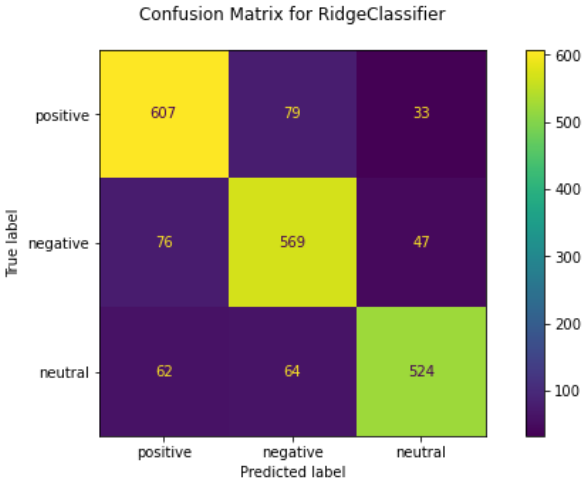

Figure 5. Confusion matrix for Ridge Classifier.

Out of the 2061 samples allocated for model testing, representing 30% of the total dataset, it is observed that the Ridge Classifier exhibits higher precision for positive sentiment polarity. Conversely, the class posing the greatest difficulty in prediction is neutral. According to the data from the confusion matrix, when an observation or Tweet is not correctly classified as neutral, it tends to be associated with a negative sentiment. Nevertheless, overall, the outlook is notably positive based on the evaluated metrics, surpassing expectations for results with supervised techniques compared to a random model or the results obtained in (Olivares et al., 2023) with unsupervised techniques.

6 Conclusions and Directions for Further Research

The sentiment polarity classification models developed achieved the highest performance with the Ridge Classifier. This algorithm is a favorable choice for sentiment analysis tasks due to several reasons. It introduces a regularization term to the linear regression equation, preventing overfitting, which is beneficial for generalizing well to new, unseen data. Additionally, the Ridge Classifier efficiently handles high-dimensional data, making it suitable for large number features scenarios, such as NLP tasks.

When comparing the results obtained in (Khan et al., 2024), where Random Forest achieved an accuracy of 0.78564 and SVM achieved an accuracy of 0.80394, the results from this proposal in the Table 2 are slightly higher. However, the overall performance of both algorithms is very similar in both works. It is important to note that there are different conditions in the datasets, but the comparison is relevant as both proposals address the same type of problem.

Comparing the results obtained in this work with the Ridge classifier to those reported in (Yadav & Ahuja, 2021), an Accuracy slightly higher than 82.5% is achieved compared to 82.29% in (Yadav & Ahuja, 2021). Similarly, comparing the F1 Score results shows .825 in this study versus .81 in (Yadav & Ahuja, 2021). The F1 Score indicates a better balance between precision and recall in the classification of the model proposed in this work.

Despite achieving better metric values through the explored supervised techniques in this work, it is crucial to emphasize that the methodology applied to this specific dataset, being supervised and restricted to learning from characteristics or patterns identified in the training set, results in predictions where terms like artificial, learn, machine, AI, and data are strongly positively associated with the negative class or polarity. It's important to note that, contextually, these terms may have greater validity if associated with a neutral polarity, implying that they do not necessarily contribute a positive or negative sentiment to a sentence.

Considering the identified problem with supervised techniques, it has been determined that employing algorithms like Lbl2Vec used in (Olivares et al., 2023), which operate in a semi-supervised manner through the definition of keywords, is the ideal approach for this sentiment analysis problem. This approach enables the control of associations between specific words and each sentiment polarity based on a given context.

References

Anrani, Y., & Lazaar, M. (2018). Random forest and support vector machine-based hybrid approach to sentiment analysis. *Elsevier*.

Embedding Projector - Visualization of High-Dimensional Data. (n.d.). TensorFlow.org. Retrieved August 21, 2022, from https://projector.tensorflow.org/.

Géron, A. (2017). *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Techniques and Tools to Build Learning Machines*. O'Reilly Media.

Khan, T. A., Sadiq, R., Shahid, Z., Alam, M. M., & Su'ud, M. M. (2024). Sentiment analysis using support vector machine and random forest. *Journal of Informatics and Web Engineering, 3*(1), 67–75. https://doi.org/10.33093/jiwe.2024.3.1.5

Olivares, J. L. (2022). Estrategias para el análisis de sentimientos de textos extraídos de Twitter utilizando técnicas de aprendizaje profundo [Tesis de licenciatura, Facultad de Ciencias de la Computación, BUAP].

Olivares, J., Sánchez, L. A., González, V. R., Santiago, D., & Zenteno, A. V. (2023). Técnicas de aprendizaje profundo. *Abstraction & Application*, 42, 124–133.

Scikit-learn. (n.d.). Retrieved February 18, 2024, from https://scikit-learn.org/stable/.

Wang, C. K. (2023). Sentiment analysis using support vector machines, neural networks, and random forests. In *Advances in Computer Science Research* (pp. 23–34). https://doi.org/10.2991/978-94-6463-300-9_4

Wartler, T. (2017, July 24). *Text Mining in Practice with R* (1st ed.). Wiley.

Yadav, A., & Ahuja, R. (2021). Comparison of parameters of sentiment analysis using different classifiers. In Gunjan, V. K., Suganthan, P. N., Haase, J., & Kumar, A. (Eds.), *Cybernetics, Cognition and Machine Learning Applications*. Algorithms for Intelligent Systems. Springer, Singapore. https://doi.org/10.1007/978-981-33-6691-6_44