Educational Administration: Theory and Practice
2024, 30(4), 7188 - 7199
ISSN: 2148-2403
https://kuey.net/

Research Article

Optimizing Feature Selection Enhancing Sentiment Analysis With Fxtend Algorithm

Mr. M. Subramaniakumar[1*], Dr. D. Maheswari[2]

[1*]Research Scholar, Rathnavel Subramaniam College of Arts and Science, Sulur
[2]Head & Research Coordinator Rathnavel Subramaniam College of Arts and Science, Sulur

Citation: Mr. M. Subramaniakumar, Dr. D. Maheswari, (2024), Optimizing Feature Selection Enhancing Sentiment Analysis With Fxtend Algorithm, *Educational Administration: Theory and Practice*, *3*(4), 7188- 7199
Doi: 10.53555/kuey.v30i4.2532

ARTICLE INFO	ABSTRACT
	Student sentiment analysis is the process of analyzing the feelings, opinions, and attitudes of students towards various aspects of their educational experience. This study proposes a feature selection method utilizing the FXtend algorithm to enhance sentiment analysis tasks. The approach involves employing Recursive Feature Elimination (RFE) with three distinct classifiers: ElasticNet, Extra Trees Classifier, and Gradient Boosting Classifier. Through iterative elimination, less relevant features are identified, aiming to retain the most informative ones for sentiment analysis. Subsequently, sentiment scores are assigned to each token in pre-processed text based on the selected features, Parts of Speech (PoS) tags, and the presence of opinion words. Aggregating these scores provides an overall sentiment assessment for the text or document. Finally, sentiment scores are normalized to a standardized scale, facilitating better interpretability and comparison across texts. This methodology promises improved accuracy and efficiency in sentiment analysis tasks, aiding in extracting meaningful insights from textual data.
	Keywords: ElasticNet, Fxtend Algorithm, Gradient Boosting Classifier, Student Sentiment Analysis

I. Introduction

Sentiment analysis, a pivotal aspect of natural language processing, plays a crucial role in deciphering the underlying emotions and opinions embedded in textual data [1]. As the field evolves, addressing the nuances of sentiment variation becomes paramount for accurate and nuanced analyses [2]. This paper introduces a novel methodology, referred to as Sentiment Variation Analysis through dataset normalization (SentiVarLSTM) [3]. By amalgamating advanced techniques such as tokenization, lowercasing, stop words removal, and PoS tagging using LSTM neural networks, SentiVarLSTM aims to provide a comprehensive solution to sentiment analysis [4-6]. The key innovation lies in the subsequent steps: the calculation of weights for PoS tags, extraction of opinion words through WordNet, and the assignment of sentiment scores to tokens [7-9]. Importantly, the normalization of sentiment scores to a common scale enhances the interpretability and comparability of results. Through this innovative approach, SentiVarLSTM endeavors to advance the field of sentiment analysis by effectively addressing variation in sentiment across diverse datasets [10-13].
In the realm of educational institutions, understanding student sentiment plays a pivotal role in fostering a conducive learning environment and enhancing educational outcomes [14]. Sentiment analysis, a burgeoning field in natural language processing, offers a powerful tool to decipher the attitudes, emotions, and opinions expressed by students in various educational contexts [15]. By analyzing textual data such as student feedback, forum discussions, or social media posts, sentiment analysis enables educators and administrators to gain valuable insights into student experiences, satisfaction levels, and potential areas for improvement [16]. However, effectively harnessing sentiment analysis in an educational setting requires careful consideration of several factors [17]. One crucial aspect is the selection of relevant features to accurately capture the nuances of student sentiment. Traditional methods often rely on manual feature engineering or simplistic approaches, which can overlook subtle cues and fail to provide comprehensive insights [18].
To address this challenge, this study introduces a novel approach to student sentiment analysis by leveraging the FXtend algorithm for feature selection [19]. The FXtend algorithm offers a sophisticated framework for

Recursive Feature Elimination (RFE), enabling the identification of the most informative features while discarding less relevant ones [20]. By applying RFE with ElasticNet, ExtraTreesClassifier, and GradientBoostingClassifier, our methodology aims to enhance the accuracy and efficiency of sentiment analysis in educational contexts [21]. Furthermore, this study goes beyond mere sentiment classification by proposing a comprehensive framework for sentiment scoring and normalization. By assigning sentiment scores to individual tokens based on selected features, Parts of Speech (PoS) tags, and the presence of opinion words, our approach facilitates a nuanced understanding of student sentiment at a granular level [22]. These scores can then be aggregated to derive an overall sentiment assessment for specific texts or documents, providing educators with actionable insights to tailor their interventions effectively [23].

1.1 Motivation of the paper

Understanding student sentiment is paramount for educational institutions striving to foster positive learning environments and enhance educational outcomes. By leveraging advanced techniques in sentiment analysis, such as feature selection with the FXtend algorithm, this study aims to provide educators and institutions with actionable insights derived from textual data sources. The proposed methodology promises to streamline sentiment analysis tasks, offering improved accuracy and efficiency in identifying student feelings, opinions, and attitudes. Ultimately, this research endeavors to empower educators to make informed decisions that positively impact student satisfaction, engagement, and overall learning experiences.

II. Background study

Dake, D.K. and Gyimah, E., (4) for sentiment analytics, this study used a dataset of qualitative student comments from Winnebago University's School of Education. The dataset was preprocessed before four supervised machine learning techniques were used to generate the model for deployment and prediction. Findings reveal that the Support Vector Machine (SVM) achieves its maximum accuracy of 63.79% with k=10, after the implementation of k-fold cross-validation for both k=5 and 3. During training and testing, the model refrained from removing stop words and stemming in order to accurately replicate the student's response and avoid any misrepresentation. The trained model achieves an impressive 92% accuracy in the practical prediction of 31 text occurrences.

Judijanto, et al. (6) Technology has become an integral part of education students' everyday lives, according to the report. This is a reflection of how much of an impact technology has had on their learning. The vast majority of people who took the survey think that technology has the ability to make learning more personalized and that it can inspire pupils to work harder. Furthermore, respondents' opinions were skewed toward supporting changes to curricula that reflect technological advancements and toward the role of educators in incorporating technology into the classroom.

Kumar, et al. (8) there has been a lot of recent and active research on sentiment analysis on brief text. Problems with formal language, misspellings, and word compression that causes high dimensionality and sparseness are only a few of the issues that require fixing in brief text. In this research, offers a new, straightforward, and successful feature selection strategy that relies on regularly distributed class-related characteristics to tackle these issues.

Sivakumar, M., & Reddy, U. S. (16) used R tools to get student comments from the Twitter API. Then, used k-mean clustering and the naïve bays classification method to analyze the sentiment. To determine the polarity of the phrases uses the sentiment package in R. determined the degree of semantic relationship between the opinion phrase and one aspect word. Based on these findings, an aspect word was assigned to each phrase. Precision, recall, and F-score were all areas where did well. It is possible to enhance the preprocessing in the future to provide more precise sentiment analysis findings for student feedback.

Srikanth, A., & Krishna, S. G. (17) In order to determine how people feel about certain passages and texts in the education database, some emotion analysis approaches use public opinion dictionaries. The terms in a sentimental and emotion dictionary are organized according to the feelings they evoke and the direction in which they are most often used. One numerical way to convey the polarity and intensity of a word or phrase is via its semantic orientation. The polarity of the communication can be determined by adding the orientation value of opinion words to the total. Additionally, they have been helpful in unsupervised categorization systems for feature extraction.

Tamrakar, M. L. (19) this research proposes an SFD sentiment classification technique that can use students' WBLMS input to determine if their sentiment is positive or negative. A raw dataset is formed by the textual WBLMS feedback that has been gathered. Both the positive and negative aspects of the WBLMS are highlighted by these remarks. The first step is to pre-process the obtained raw SFD dataset. Bow and TD-IDF, two feature extraction methods, were then used to transform the raw text into feature vectors. For SFD datasets, the Bag of Words (BOW) transforms the text document collection into a matrix of feature vector counts, which provides the number of times a word occurs. To illustrate the importance of a word to a document, the TF-IDF approach is used. The comments are categorized as either positive or negative using the SFD dataset.

Yan, W.,et al. (23) Every day, social media platforms produce vast quantities of user-generated content due to the proliferation of mobile Internet and the widespread commercial usage of 5G technology. Accurately gauging people's opinions and emotional dispositions is now a pressing issue that needs fixing. In light of this, it provides a CNN-BiGRU-AT model for sentiment analysis in student-written texts. Word characteristics are obtained using CNN, sentence features using BiGRU, and text features using AT, in that order. In order to finish classifying students' emotions, more characteristics are provided to the soft max classifier via the top-down analysis of phrase and word associations.

Table 1: Comparison table for existing work

Author	Year	Methodology	Limitations	Advantages
Anam, M. K.et al.	**2020**	K-NN	There is potential for K-NN to enhance sentiment categorization, as shown by its 56% accuracy.	Provides helpful information on student feelings, which helps in making educated judgments in the classroom.
Dake, D.K. and Gyimah, E.,	**2023**	Machine Learning	The efficacy of chosen machine learning algorithms varies among datasets, which in turn affects performance.	The use of text analytics in the classroom helps advance the SMART framework for school buildings.
Judijanto,et al.	**2023**	Artificial Intelligence	This study only included 20 students; therefore the results cannot be applied to a larger population.	Learning experiences are greatly enhanced by AI technology, particularly VR and AR.
Kumar,et al.	**2018**	Sentiment Analysis	Limitations on generalizability can exist due to the fact that effectiveness is affected by unique dataset features.	Introduces a new approach to feature selection that overcomes the obstacles of short-text sentiment analysis and outperforms the current state of the art.
Okoye, et al.	**2022**	Machine Learning	The performance of the model is dependent on the classifier in question, and its relevance to different classifiers can differ.	In order to get valuable information from assessments of teachers, EPDM+ML presents a new combination of educational process, data mining, and machine learning.

2.1 problem definition

This study addresses the need to effectively analyze student sentiment, aiming to extract valuable insights from textual data sources to enhance educational experiences and outcomes. By proposing a feature selection method utilizing the FXtend algorithm, the research seeks to streamline sentiment analysis tasks, ultimately aiding educators in understanding student feelings, opinions, and attitudes towards their educational experiences.

III. Materials and methods

This section outlines the materials and methodologies employed in the study to conduct sentiment analysis on student data. The detail the datasets used, the feature selection approach utilizing the FXtend algorithm, and the sentiment analysis methodology including sentiment scoring and normalization techniques. Through a combination of advanced feature selection and sentiment analysis methodologies, the study aim to provide a robust framework for understanding student sentiment in educational contexts.

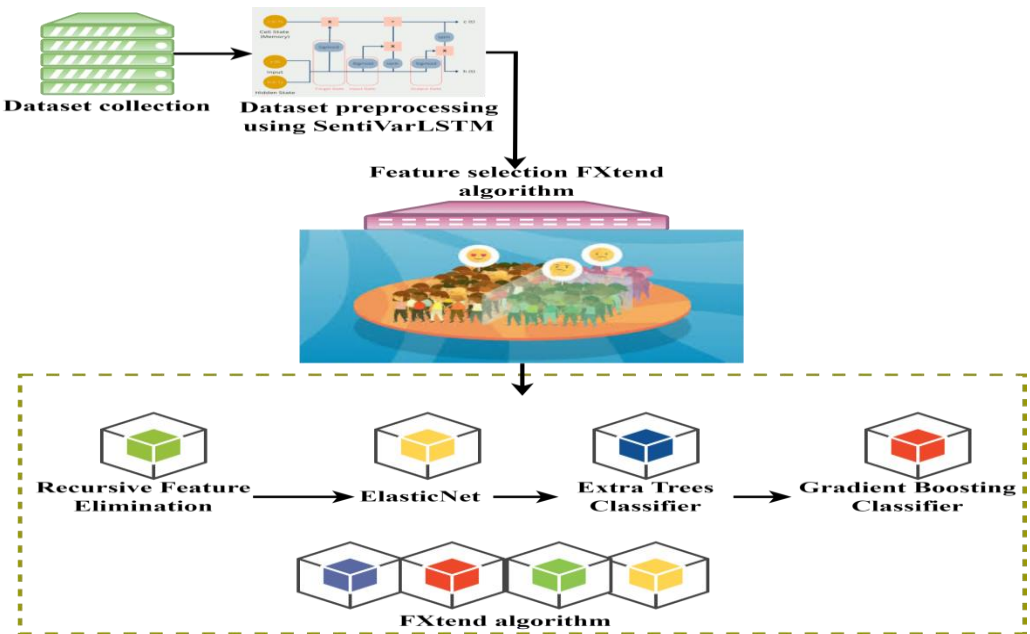

Figure 1: Overall workflow architecture

3.1 Dataset collection

In this section, this research utilizes three different educational datasets for student sentiment analysis. The first dataset, collected from the Kaggle website (https://www.kaggle.com /datasets/jayaprakashpondy/ student-feedback), comprises a CSV file with a size of 4.28MB. The dataset is distributed across four files, with the primary data stored in "feedback_student.csv." This file consists of three attributes and a total of 2346 records. The second dataset, obtained from Kaggle as well (https:// www.kaggle.com /datasets/brarajit18/student-feedback-dataset), is in XLSX format and has a size of 37KB. This dataset is contained within a single file and includes 12 attributes with a total of 186 records. The third dataset was obtained from ExportComments as well and can be accessed through the following link: https: //exportcomments.com/done/b273e2b8-b359-4d66-9a11-6d7f05527904. We collected the educational video links from YouTube for analysis purposes.

3.2 Dataset preprocessing using SentiVarLSTM

Student Sentiment Analysis using SentiVarLSTM involves a multi-step approach to deciphering sentiment nuances in textual data. The process begins with dataset pre-processing, including tokenization, lowercasing, stop words removal, and text cleaning to ensure consistency and eliminate irrelevant information. The unique contribution of SentiVarLSTM lies in its application of a LSTM neural network for PoS tagging during this pre-processing phase, capturing intricate contextual information. Subsequently, the model calculates weights for PoS tags based on their relevance in sentiment determination. Leveraging WordNet, a lexical database, opinion words are identified, enriching the understanding of sentiment expressions. Sentiment scores are then assigned to each token, incorporating both PoS tag weights and the presence of opinion words. Notably, SentiVarLSTM's architecture spans 19 layers, augmenting its capacity to comprehend and analyze complex sentiment variations. By virtue of its comprehensive dataset normalization and sophisticated methodology, SentiVarLSTM emerges as a robust framework for sentiment analysis, rendering it invaluable for deciphering and interpreting sentiments across a spectrum of textual contexts.

$\mathrm{c_t} = \mathrm{f_t} * \mathrm{C_{t-1}} + \mathrm{i_t} * \mathrm{c_t}$ ---- (1)

$\mathrm{h_t} = \mathrm{o_t} * \mathrm{GELU}\ \mathrm{c_t}$ -------- (2)

As a function of the coordinates of its whole output, the LSTM value of the cascaded structures is represented by $Yns = [hr - n, \ldots, hr - i]$. It is just necessary to estimate the last output sequence attribute, hr-i, while thinking about problems with overall performance assessment. The formula used to calculate SentivarLSTM involves a multi-step process: beginning with tokenization and preprocessing of input text to extract individual tokens and clean irrelevant information. Following this, Part-of-Speech (PoS) tagging assigns grammatical categories to tokens, crucial for contextual understanding. Weighting PoS tags based on their relevance in sentiment determination is conducted next, followed by identification of opinion words using resources like WordNet. Sentiment scores are then assigned to tokens, considering both PoS tag weights and the presence of opinion words.

3.3 Feature selection using FXtend algorithm

The feature selection process is a critical step in sentiment analysis, aiming to identify the most informative features while eliminating irrelevant ones. This study employs the FXtend algorithm, a powerful tool for Recursive Feature Elimination (RFE), to streamline this process. The FXtend algorithm iteratively selects features based on their importance in classifying sentiment, enhancing the efficiency and accuracy of sentiment analysis tasks. It operates by training a classifier on the full feature set and subsequently ranking the features based on their importance scores. During each iteration, the algorithm eliminates the least important features and re-evaluates the classifier's performance. This process continues until the desired number of features is reached or until performance metrics plateau. By iteratively eliminating less relevant features, FXtend identifies a subset of features that maximize sentiment classification accuracy while minimizing computational overhead.

3.3.1 Recursive Feature Elimination

One feature selection strategy used in machine learning is recursive feature elimination (RFE). Iteratively, RFE removes the least significant features from the feature set until the target number is obtained. Before determining the relative value of each feature, a classifier is trained on the whole collection of features referred by Kumar et al. (2018). The classifier then supplies a ranking measure for this purpose. After that, remove the characteristics that aren't crucial and retrain the classifier using that smaller set of features. This technique is iterated until the number of features maintained is the required value or until some stopping requirement is satisfied. Particularly helpful for high-dimensional datasets, RFE improves model performance by decreasing overfitting while maintaining the most essential features. This research, uses the FXtend algorithm to execute.

$Rank_i = \{r_{i1} = 1, r_{i2} = 2, \ldots, r_{ip} = p\}$ --------- (3)

Next, ranking the features using eight different ML-RFE algorithms and find their cut-off points. For the purpose of selecting the feature with a rank larger than or equal to α ∈(0, 1), ranking all features in a certain way. These characteristics will be considered crucial by the majority. The individual optimum feature subset is constructed by selecting ⌊αp⌋ features from each feature subset in this manner.

$FS_i^{opt} = \{f_{i1}, f_{i2}, \ldots, f_i|ap|\}$ ------------ (4)

$fj^{opt} = \{FS_1^{opt}, FS_2^{opt}, \ldots, FS_N^{opt}\}$ -------------- (5)

It is reframed to the robust biomarker screening problem as an N-feature subset stable combination challenge. Every conceivable combination of the sets in FSopt has its stability calculated. The eight ML-RFE techniques determine the number of feature subsets that can be combined from any two subsets C2N, which can be further combined to form CN~1N. As a result, in all, there are ΓN~1k=2N!k!(N~k)!.mix of all possible values for N≥3.

As a last step, finding the ultimate target feature set is indicated by the combination of FSopt attaining the greatest stability value. As the screened robust biomarkers, the study choose the characteristics in the combinations with the highest stability values based on the concept of who occurs more often, which means greater and equal to the fixed parameter κ,

$Where 1 \leq i \leq N, 1 \leq j \leq |ap| and 1 \leq m \leq \sum_{k=2}^{N-1} \frac{N!}{k!(N-k)!}$ ------------ (6)

3.3.2 ElasticNet

ElasticNet is a linear regression model that integrates the regularization techniques of Lasso (L1 regularization) and Ridge (L2 regularization) regression, effectively addressing their individual limitations. It is well-suited for high-dimensional datasets where the number of features surpasses the number of samples. By combining the penalties of Lasso and Ridge, ElasticNet offers a balanced approach to variable selection and multicollinearity reduction referred by Okoye et al. (2022). The model optimizes a cost function comprising the sum of squared errors and penalizes both the absolute and squared coefficients of features. The regularization strength is governed by two hyperparameters: alpha, determining the overall regularization intensity, and l1_ratio, controlling the balance between L1 and L2 penalties. With its versatility and effectiveness in handling correlated predictors, ElasticNet finds widespread application in tasks like feature selection, regression, and classification within the realm of machine learning.

All of the forecasting models are constructed using Elastic Net. Since Elastic Net relies on least squares regression, regularization—most often, the lasso and ridge algorithms—are typically used to prevent the model from being over fit. When extracting sparse features, Lasso linear regression use L1 regularization, which makes it simple to lose the original information; in contrast, ridge linear regression employs L2 regularization, which takes more time and has a regularization coefficient that decays too slowly. An enhanced version of the lasso and ridge linear regression algorithms, Elastic Net strikes a good compromise between models sparsely and training speed. In the following expression, the linear regression issue is described: Y represents the output sequence, X the input sequence, μ is the parameter matrix, and ε is a random error that follows a normal distribution.

$Y = X^T\mu + \varepsilon$ ----------- (7)

In the Elastic Net construction. τThe target parameter matrix with estimate is denoted by

$$\begin{cases} L(\lambda_1, \lambda_2, \mu) = \|Y - X^T\mu\| + \lambda_2\|\mu\|^2 + \lambda_1\|\mu\|_1 \\ \tau = \arg\min_{\beta}\{L(\lambda_1, \lambda_2, \mu)\} \end{cases}$$ --------------- (8)

After that, you should proceed to tackle the following problem:

$\tau = \arg\min_{\mu}\|Y - X^T\mu\|^2,$ -------------- (9)

$subject\ to\ (1-\alpha)\|\mu\|_1 + \alpha\|\mu\|^2 \leq t$ --------------- (10)

Term, which is a convex mixture of lasso and ridge penalty terms. At 1, Elastic Net is the same as lasso regression, at 0, it's the same as ridge regression, and as t approaches infinity, Elastic Net is the same as simple least squares regression. In comparison to AI algorithms, Elastic Net is more suited for real time prediction of large-scale load data since it reduces model training time and resource consumption without sacrificing forecast accuracy. The t parameter constrains the $(1-\alpha)\|\mu\|_1 + \alpha\|\mu\|^2$.

r are the error values of MAPE and RMSE, respectively. δ_mAnd δ_r and the regularization range coefficient t are the primary parameters. The grid search approach is used to automatically search the model parameters associated with each kind of load within the artificially provided parameter range, guaranteeing that the forecasting model is effective for each load type.

$$\begin{cases} \delta_m = \frac{1}{N}\sum_{i=1}^{N}\frac{|\hat{y}_i - y_i|}{y_i} \times 100\% \\ \delta_r = \sqrt{\frac{\sum_{i=1}^{N}(\hat{y}_i - y_i)^2}{N}} \end{cases}$$ --------- (11)

3.3.3 Extra Tree Classification

Extremely Randomised Trees Classification, or Extra Trees Classification, is a decision tree-based ensemble learning approach. During training, it constructs a number of decision trees and then averages their predictions to make the model more accurate and resilient referred by Sivakumar, M., & Reddy, U. S. (2017). However, what sets Extra Trees apart is its randomness in selecting candidate splits. Unlike traditional decision trees that search for the best split at each node based on a subset of features, Extra Trees randomly select splits, leading to a higher level of diversity among the trees. Additionally, Extra Trees tend to be computationally efficient because they require less computation to find the best split. Overall, Extra Trees Classification is a powerful algorithm for classification tasks, particularly suitable for high-dimensional datasets with noisy or correlated features.

Decision tree-based ensemble learning approaches like random forests include an extension called Extra Tree Classifiers, which is also called much randomized trees. An enhancement to the random forest technique, Extra Tree Classifiers constructs a network of decision trees using more randomization during the node-splitting phase. The method of splitting at each node is the main distinction between Extra Tree Classifiers and conventional random forests. A random forest chooses the optimal split by selecting and evaluating a subset of characteristics. On the other hand, Extra Tree Classifiers pick the optimal split from among randomly chosen points for each subgroup characteristic. The result of this additional randomness in the split selection process is an ensemble of trees that is both more diversified and less correlated. By combining the forecasts from each decision tree, the Extra Tree Classifier arrives at its final prediction. Using a majority vote to decide on the final class prediction is the most typical method. In Figure 6, see the visual depiction. Therefore, for an N-size learning sample l_{SN}

$l_{S_N} = \{(x^i, y^i): i = 1, \dots, N\}$ ------------ (12)

$x_j^{(0)} = -\infty \ \ and \ \ x_j^{(N+1)} = +\infty \ \ \forall j = 1, \dots, N$ ---------------- (13)

and denote ∀(i1, . . . ,in) ∈ 0, . . . , N by I(i1,...,in)(x). Hence, the characteristic function of the hyper-interval can be represented as

$\left[x_1^{(i_1)}, x_1^{(i_1+1)}\right] \times \cdots \times \left[x_n^{(i_1)}, x_n^{(i_1+1)}\right].$ ----------------- (14)

$\hat{y}(x) = \sum_{i_1=0}^{N} \cdots \sum_{i_1=0}^{N} I_{(i_1,\dots,i_n)}(x) \ \ \sum_{X \cup \{x_1,\dots,x_n\}} \lambda_{(i_1,\dots,i_n)}^{X} \ \ \ \prod_{x_j \in X} x_j$ -------- (15)

3.3.4 Gradient boosting classifier

One effective ensemble learning method for classification applications is the Gradient Boosting Classifier. It achieves its results by gradually incorporating weak learners, often decision trees, into the ensemble, with each successive tree addressing the shortcomings of its predecessors referred by Wu, Y. et al. (2023). The key idea behind gradient boosting is to optimize a loss function by iteratively minimizing the errors of the ensemble. During training, the model starts with an initial prediction (usually the mean of the target variable for regression or a constant for classification) and then fits a new decision tree to the residuals (the differences between the predicted and actual values). Subsequent trees are trained to predict the residuals of the ensemble formed by the previous trees. This iterative process continues until a predefined number of trees is reached, or until a stopping criterion is met. When it comes to non-linear correlations between features and target variables, gradient boosting classifiers are masters at managing complicated datasets. They perform very well in real-world applications and machine learning contests, and they are resistant to overfitting.

An enhanced version of GBDT, Gradient boosting is a tree boosting technique. By adding up the scores of all the decision trees in a sample space D with n samples and m features, we can get a tree boosting model:

$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i), f_k \in F$ --------- (16)

Along with the following definition of the XGBoost goal function:

$Obj(\theta) = \sum_{i=1}^{n} L(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k)$ ---------- (17)

In such case $D = \{(x_i, y_i)\}(|D| = n, x_i \in R^m, y_i \in R)$, F is the space of regression trees (like CART) shown above. The number of CART, denoted as K, is a function map connecting each sample point to a fraction. (, first) A differentiable convex loss function, denoted as $L(y_i, \hat{y}_i)$, assesses the deviation from the goal y_i as well as the forecast $\hat{y}_i$. An anti-over-fitting regularization term is f_k. It is the same as minimizing the objective function while training the $t\ th$ tree:

$Obj^{(t)} = \sum_{i=1}^{n} L\left(y_i, \hat{y}_i^{(t)}\right) \ + \sum_{k=1}^{t} \Omega(f_k)$ -------- (18)

$= \sum_{i=1}^{n} L\left(y_i, \hat{y}_i^{(t1)} + \ f_t(x_i)\right) + \Omega(f_t) + const$

One unique aspect of Gradient boosting is its use of the second-order Taylor expansion of the loss function to mimic the original loss function. It may approximatively solve the goal function given above as:

$Obj^{(t)} \approx \sum_{i=1}^{n} [L\left(y_i, \hat{y}_i^{(t-1)}\right) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)]$ -------- (19)

$+\Omega(f_t) + const$

3.3.5 FXtend algorithm

The FXtend algorithm is an advanced feature selection technique that leverages Recursive Feature Elimination (RFE) in conjunction with various classifiers to enhance the accuracy and efficiency of sentiment analysis and other machine learning tasks. FXtend iteratively eliminates less relevant features from the feature set while retaining the most informative ones. It employs RFE with a combination of classifiers such as ElasticNet, Extra Trees Classifier, and Gradient Boosting Classifier, allowing for a comprehensive evaluation of feature importance. By iteratively assessing the impact of feature elimination on classifier performance, FXtend identifies a subset of features that maximize predictive accuracy while minimizing computational complexity. This iterative refinement process helps to improve model generalization and mitigate overfitting, ultimately enhancing the reliability and interpretability of the machine learning model. FXtend algorithm is particularly useful in scenarios with high-dimensional datasets where feature selection is crucial for optimizing model performance.

In the realm of student sentiment analysis, understanding the complex interplay of factors influencing student perceptions and attitudes is essential. Here, the study set aside considerations such as survey biases and data collection intricacies, focusing solely on extracting insights from students' expressed sentiments. The study examines the sentiments expressed by students towards various aspects of their educational experience, represented as sentiment scores denoted as S.

$r_{\$€}, r_{\$£}, r_{€£}$ --------- (20)

The sentiment scores for each aspect can be denoted as $r_{\$£}$, $r_{£€}$, and $r_{\$€}$, respectively. These sentiment scores represent the subjective feelings, opinions, and attitudes of students towards each aspect, ranging from negative to positive sentiments.

$r_{\$£} \cdot r_{£€} > r_{\$€} \cdot$ ------------ (21)

$r_{\$€}^{(new)} = \frac{r_{\$£}}{r_{€£}}, \; r_{\$£}, \quad r_{€£} \cdot$ ----------- (22)

Now, let's introduce the concept of sentiment arbitrage, where students can adjust their sentiments towards different aspects of their educational experience to capitalize on perceived discrepancies or opportunities for improvement. This adjustment in sentiment can be represented mathematically as follows:

$r_{€\$} \cdot r_{\$£} > r_{€£} \cdot$ --------------- (23)

$r_{\$€}, \quad r_{\$£}, \quad r_{€£}^{(new)} = \frac{r_{\$£}}{r_{\$€}}$ ---------- (24)

$r_{£€} \cdot r_{€\$} > r_{£\$}$ ------------- (25)

These equations illustrate how students can strategically adjust their sentiments towards different aspects of their educational experience to exploit arbitrage opportunities and achieve a more favorable sentiment landscape.

$r_{\$€}, \quad r_{\$£}^{(new)} = r_{\$€} \cdot r_{€£}, \quad r_{€£} \cdot$ ------------ (26)

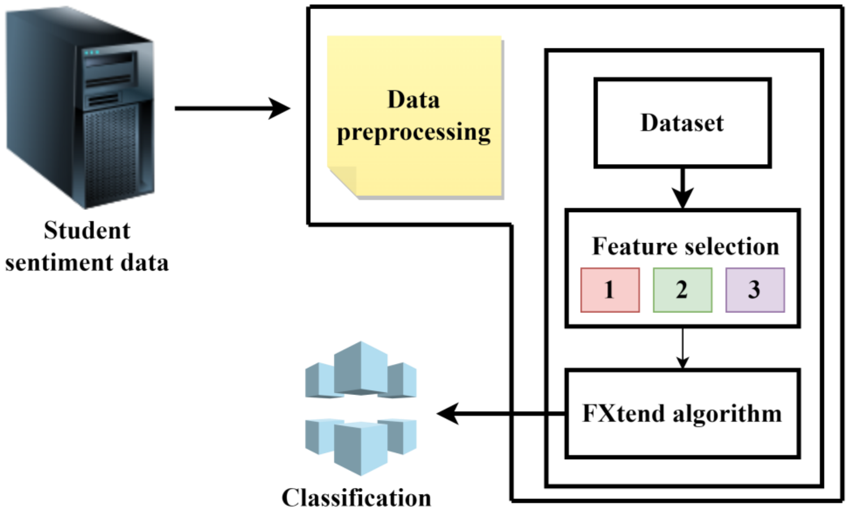

Figure 2: FXtend algorithm architecture

Algorithm 1: FXtend algorithm

Input:
- Dataset containing sentiment scores ($r_\$£, r_£€, r_\$€$) for each aspect of the educational experience.

Steps:
1. **Initialization**: Start with the initial sentiment scores for each aspect of the educational experience ($r_\$£, r_£€, r_\$€$).
2. **Identify Arbitrage Opportunities**: Compare the sentiment scores for different aspects of the educational experience to identify potential arbitrage opportunities using equations (17) to (21).
3. **Execute Sentiment Adjustments**:
   - If conditions for arbitrage are met, adjust sentiment scores to exploit the arbitrage opportunity.
   - For example, if $r_\$£ \cdot r_£€ > r_\$€$, adjust sentiment scores for aspect 1 and aspect 2 to exploit the arbitrage opportunity.
   - Similarly, adjust sentiment scores for other aspects based on the conditions defined in equations (19) to (21).
4. **Update Sentiment Scores**: After each adjustment, update the sentiment scores accordingly to reflect the changes made.

Output:
- Identification of arbitrage opportunities based on the sentiment scores.

IV. Results and discussion

This section presents the findings of our study on sentiment analysis in educational contexts, following the feature selection process using the FXtend algorithm. The research delves into the performance metrics of the selected features and discusses their implications for understanding student sentiment. Through a comprehensive analysis of the results, the study aim to uncover insights that can inform educational practices and enhance student experiences.

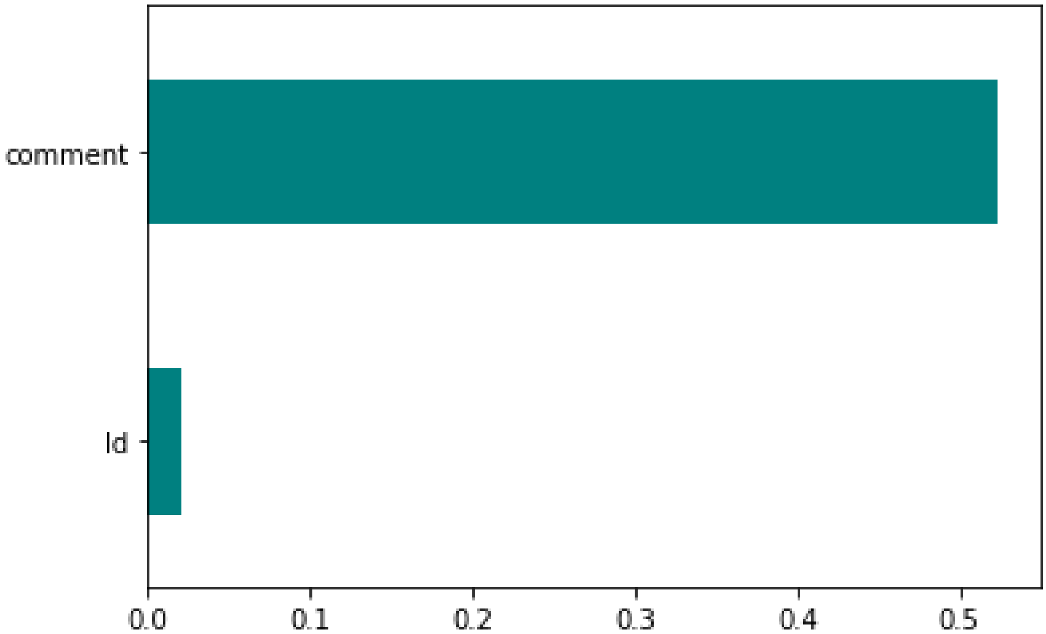

Figure 3: Important feature selection criteria

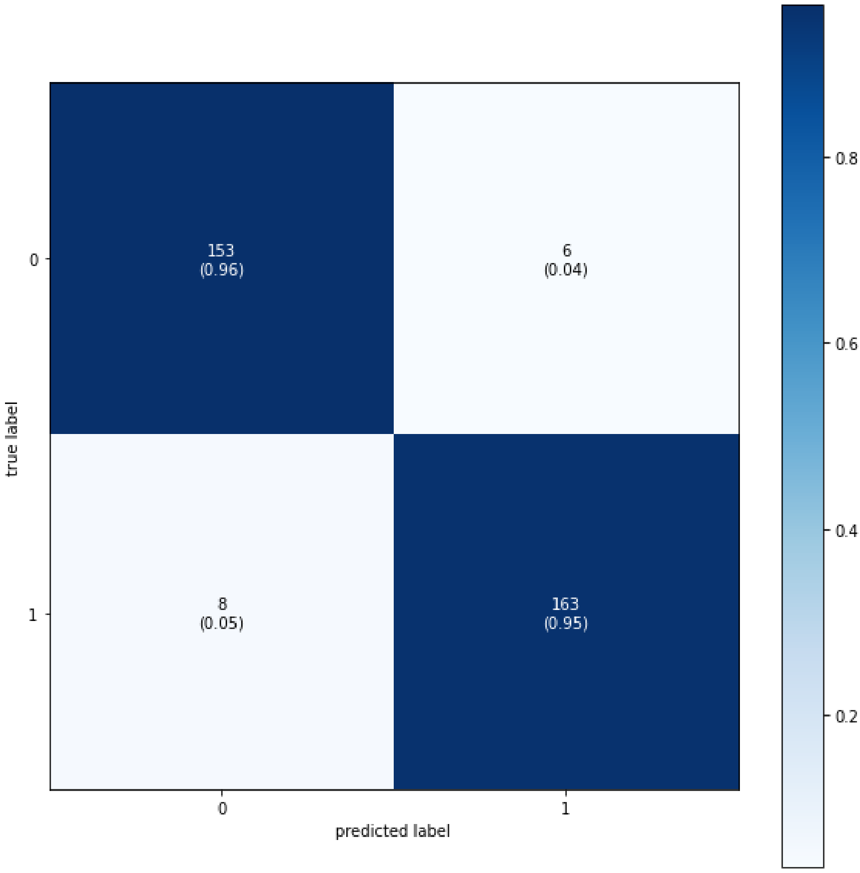

Figure 4: Confusion matrices

4.1 Performance evaluation

1. Accuracy: The fraction of samples with the right classification out of all samples. Mathematically:

$$Accuracy = \frac{(TP + TN)}{(TP + FP + TN + FN)} \text{ ----------- } (27)$$

2. Precision: Ratio of student sentiment analysis samples with accurate identification to total student sentiment analysis samples with accurate identification. Mathematically:

$$Precision = \frac{TP}{TP + FP} \text{ ------------ } (28)$$

3. Recall (also known as sensitivity or true positive rate): The proportion of correctly classified student sentiment analysis samples out of the total number of actual student sentiment analysis samples. Mathematically:

$$Recall = \frac{TP}{TP + FN} \text{ -------------- } (29)$$

4. F1 score: A middle ground between accuracy and memory that strikes a harmonic mean. Mathematically:

$$F1\, score = 2 * Precision * Recall / (Precision + Recall) \text{ --------- } (30)$$

Table 2: Classification performance metrics comparison

	methods	Accuracy	Precision	Recall	F-measure
Existing method	DT	91.68	93.34	95.31	94.11
	SVM	92.01	94.21	96.68	95.24
	LR	92.11	95.61	96.95	96.25
	GNB	93.31	94.38	95.91	95.36
Proposed method	FXtend	94.24	95.68	96.36	97.31

The table 2 shows comparison of performance metrics between existing methods (Decision Tree, Support Vector Machine, Logistic Regression, Gaussian Naive Bayes) and the proposed method (FXtend) reveals notable distinctions. While all methods exhibit high levels of accuracy, with FXtend achieving the highest accuracy at 94.24%, precision, recall, and F-measure metrics provide a more nuanced understanding of their efficacy. FXtend also outperforms existing methods in precision (95.68%), recall (96.36%), and F-measure (97.31%), indicating its superiority in correctly identifying positive instances, capturing relevant instances, and achieving a balance between precision and recall. Among existing methods, Logistic Regression demonstrates the highest precision (95.61%), recall (96.95%), and F-measure (96.25%), closely followed by Support Vector Machine, suggesting their effectiveness in classifying positive instances and capturing relevant instances. Gaussian Naive Bayes, while exhibiting respectable performance across metrics, falls slightly behind in precision (94.38%), recall (95.91%), and F-measure (95.36%) compared to other methods. These findings underscore the efficacy of FXtend in sentiment analysis tasks, promising improved accuracy and reliability in extracting meaningful insights from textual data in educational contexts.

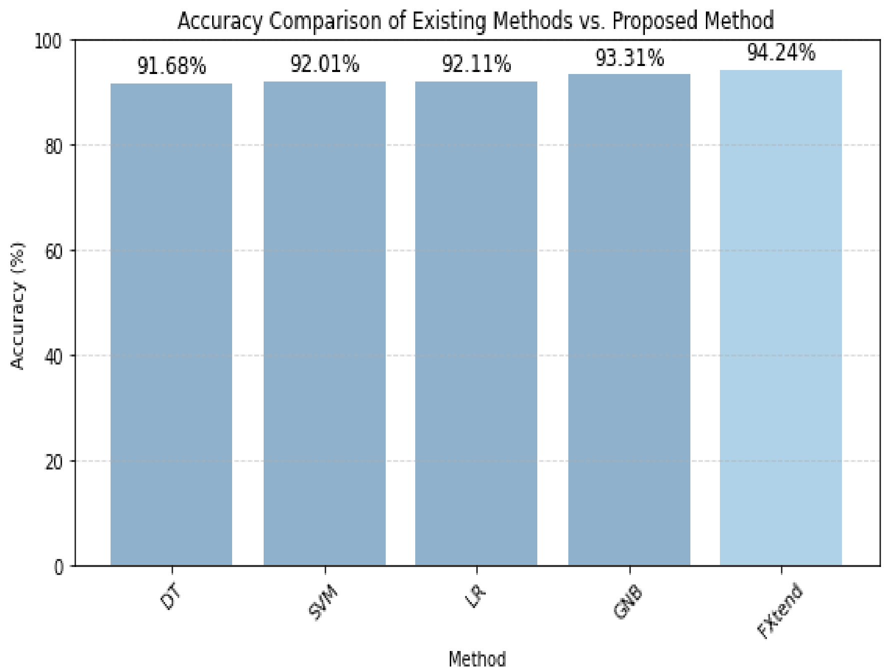

Figure 5: Accuracy comparison chart

Figure 5 displays a chart comparing accuracy. Accuracy numbers are shown on the y-axis and techniques are shown on the x-axis.

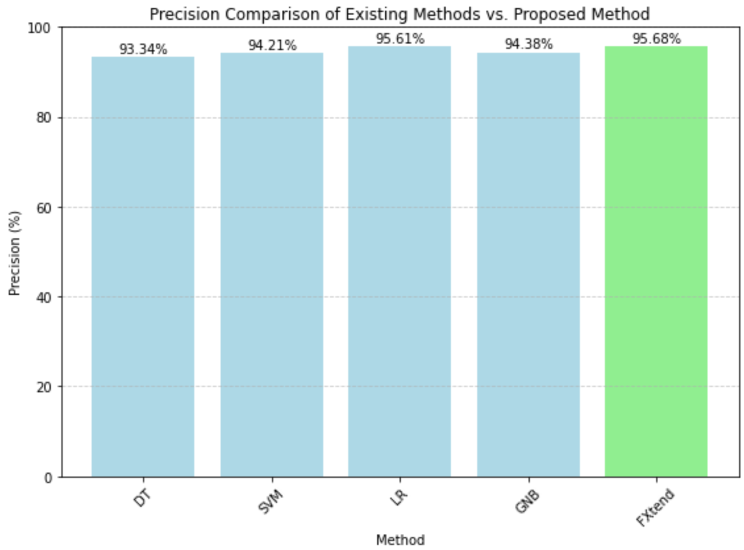

Figure 6: Precision values comparison chart

A comparison chart showing accuracy is shown in figure 6. Methods are shown on the x-axis, while accuracy values are shown on the y-axis.

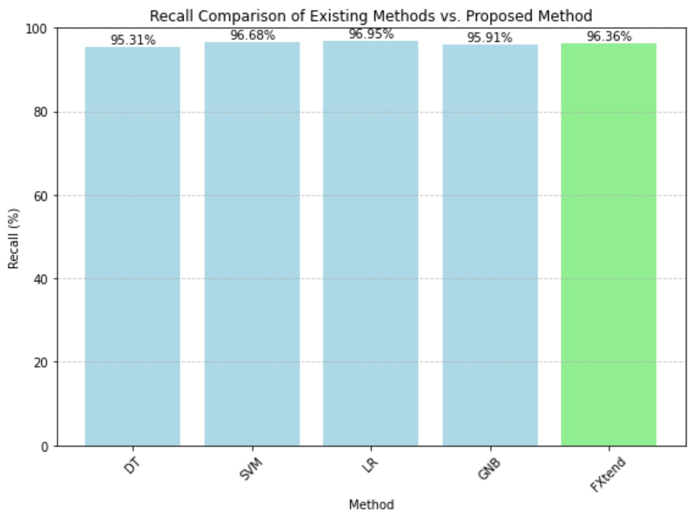

Figure 7: Recall comparison chart

Figure 7 displays a chart comparing recalls. Recall values are shown on the y-axis while procedures are shown on the x-axis.

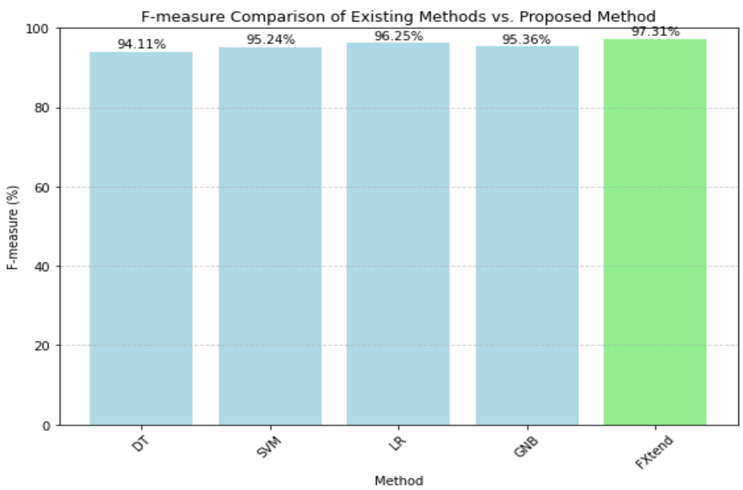

Figure 8: F-measure value comparison chart

Figure 8 displays a chart comparing F-measures. Axis points indicate procedures, whereas y-axis points reveal f-measure values.

V. Conclusion

In conclusion, our study introduces a robust feature selection method utilizing the FXtend algorithm to enhance sentiment analysis tasks in educational settings. By employing Recursive Feature Elimination (RFE) with three diverse classifiers, namely ElasticNet, Extra Trees Classifier, and Gradient Boosting Classifier, the study systematically identify the most informative features while discarding less relevant ones. Subsequently, sentiment scores are assigned to each token in pre-processed text based on the selected features, Parts of Speech (PoS) tags, and the presence of opinion words, allowing for a nuanced understanding of student sentiment. Aggregating these scores provides a comprehensive sentiment assessment for texts or documents, further augmented by the normalization of sentiment scores to a standardized scale. This methodology not only promises improved accuracy and efficiency in sentiment analysis tasks but also facilitates the extraction of meaningful insights from textual data, ultimately contributing to informed decision-making and enhanced educational outcomes.

VI. Reference

1. Anam, M. K. (2022). Sentiment Analysis of Online Lectures using K-Nearest Neighbors based on Feature Selection. Jurnal Nasional Pendidikan Teknik Informatika: JANAPATI, 11(3), 216-225.
2. Biró, Attila, Antonio Ignacio Cuesta-Vargas, and László Szilágyi. "Precognition of mental health and neurogenerative disorders using AI-parsed text and sentiment analysis." Acta Universitatis Sapientiae, Informatica 15, no. 2 (2023): 359-403.
3. Chandrasekaran, S., Dutt, V., Vyas, N., & Kumar, R. (2023, January). Student Sentiment Analysis Using Various Machine Learning Techniques. In 2023 International Conference on Artificial Intelligence and Smart Communication (AISC) (pp. 104-107). IEEE.
4. Dake, D.K. and Gyimah, E., 2023. Using sentiment analysis to evaluate qualitative students' responses. Education and Information Technologies, 28(4), pp.4629-4647.
5. Farrow, E., 2018. Automated content analysis for modelling student engagement in online discussion forums.
6. Judijanto, Loso, Aswamedhika Aswamedhika, Ismul Aksan, and Idam Mustofa. "Student Sentiment Analysis: Implementation of Artificial Intelligence in Improving Teaching Quality." Journal of Social Science Utilizing Technology 1, no. 4 (2023): 227-238.
7. Kastrati, Z., Dalipi, F., Imran, A. S., Pireva Nuci, K., & Wani, M. A. (2021). Sentiment analysis of students' feedback with NLP and deep learning: A systematic mapping study. Applied Sciences, 11(9), 3986.
8. Kumar, HM Keerthi, and B. S. Harish. "A new feature selection method for sentiment analysis in short text." Journal of Intelligent Systems 29, no. 1 (2018): 1122-1134.
9. Misuraca, M., Forciniti, A., Scepi, G., & Spano, M. (2020). Sentiment Analysis for Education with R: packages, methods and practical applications. arXiv preprint arXiv:2005.12840.
10. Mostafa, L. (2019, October). Student sentiment analysis using gamification for education context. In International conference on advanced intelligent systems and informatics (pp. 329-339). Cham: Springer International Publishing.
11. Okoye, Kingsley, Arturo Arrona-Palacios, Claudia Camacho-Zuñiga, Joaquín Alejandro Guerra Achem, Jose Escamilla, and Samira Hosseini. "Towards teaching analytics: a contextual model for analysis of students' evaluation of teaching through text mining and machine learning classification." Education and Information Technologies (2022): 1-43.
12. P. B. Sunitha, S. Joseph and P. V. Akhil, "A Study on the Performance of Supervised Algorithms for Classification in Sentiment Analysis," *TENCON 2019 - 2019 IEEE Region 10 Conference (TENCON)*, Kochi, India, 2019, pp. 1351-1356, doi: 10.1109/TENCON.2019.8929530.
13. Pong-Inwong, C., & Kaewmak, K. (2016, October). Improved sentiment analysis for teaching evaluation using feature selection and voting ensemble learning integration. In 2016 2nd IEEE international conference on computer and communications (ICCC) (pp. 1222-1225). IEEE.
14. Rahman, Shaomi, Jonayed Nafis Hemel, Syed Junayed Ahmed Anta, and Hossain Al Muhee. "Sentiment Analysis using R: An approach to correlate Bitcoin price fluctuations with change in user sentiments." PhD diss., BRAC University, 2018.
15. Ren P, Yang L, Luo F. Automatic scoring of student feedback for teaching evaluation based on aspect-level sentiment analysis. Education and information technologies. 2023 Jan;28(1):797-814.
16. Sivakumar, M., & Reddy, U. S. (2017). Aspect based sentiment analysis of students opinion using machine learning techniques. 2017 International Conference on Inventive Computing and Informatics (ICICI). doi:10.1109/icici.2017.8365231

17. SRIKANTH, A., & KRISHNA, S. G. Student Sentimental Analysis for Educational Database UsingUnsupervised Deep Learning Approaches.
18. Sultana, J., Sultana, N., Yadav, K., & AlFayez, F. (2018, April). Prediction of sentiment analysis on educational data based on deep learning approach. In 2018 21st Saudi computer society national computer conference (NCC) (pp. 1-5). IEEE.
19. Tamrakar, M. L. (2021). An Analytical Study Of Feature Extraction Techniques For Student Sentiment Analysis. Turkish Journal of Computer and Mathematics Education (TURCOMAT), 12(11), 2900-2908.
20. Watkins, Johanna, Marcos Fabielli, and Mufti Mahmud. "Sense: a student performance quantifier using sentiment analysis." In 2020 International Joint Conference on Neural Networks (IJCNN), pp. 1-6. IEEE, 2020.
21. Wu, Rong, and Zhonggen Yu. "Do AI chatbots improve students learning outcomes? Evidence from a meta-analysis." British Journal of Educational Technology 55, no. 1 (2024): 10-33.
22. Wu, Y., Ming, Z., Allen, J.K. and Mistree, F., 2023. Evaluation of Students' Learning Through Reflection on Doing Based on Sentiment Analysis. Journal of Mechanical Design, 145(3), p.032301.
23. Yan, W., Zhou, L., Qian, Z., Xiao, L. and Zhu, H., 2021. Sentiment analysis of student texts using the CNN-BiGRU-AT model. Scientific Programming, 2021, pp.1-9.