Journal of Computer Science

Research Article

Decoding Discrepancies: Understanding Rating and Review Gaps in Best-Selling Products

[1,2]Jyoti S Verma and [1]Jaimin N Undavia

[1]Faculty of Computer Science and Applications, Smt. Chandaben Mohanbhai Patel Institute of Computer Applications, Charotar University of Science & Technology, Gujarat, India
[2]Shri G.M Bilakhia College of Applied Sciences, Rajju Shroff ROFEL University, Vapi, India

Article history
Received: 29-12-2024
Revised: 26-02-2025
Accepted: 03-04-2025

Corresponding Author:
Jyoti S Verma
Faculty of Computer Science and Applications, Smt. Chandaben Mohanbhai Patel Institute of Computer Applications, Charotar University of Science & Technology, Gujarat, India;
Shri G.M Bilakhia College of Applied Sciences, Rajju Shroff ROFEL University, Vapi, India
Email: jyoti.s.vermaa@gmail.com

Abstract: Classification of the sentiment and using it for the rating prediction is the major task don in this paper. The emotions and opinions are expressed by the users about a product on the e-commerce website and this provides the basis for the prediction done here. The prediction of the rating enables both the user as well as the seller as the user will get the trust worthiness of the product quality whereas the seller will get an insight into the issues and the upgradations if any needed in the product.the reviews are easily available online for the products these reviews . Various websites like Myntra, Amazon , Flipkart and many more have reviews displayed depicting the users verdit whether to buy or not buy the product. The reviews hence collected are preprocessed using NLP algorithm like VADER, TextBlob, Flair deep learning algorithms like CNN, ANN (Lavanya & Sasikala, 2021). The reviews are filtered to classified as positive, neutral or negative based on the emotions embedded in it after collecting the sentiment score this score is used to train the model and predict the rating of the product.

Keywords: Machine Learning, Deep Learning, Sentiment Analysis, Text Mining, Data Mining, NLP, Recommender System, AdaBoost, Stacking, Hybrid, LSTM

Introduction

Humans have the priceless attribute –"Emotions," and these emotions help in decision-making and choosing the right and ignoring the wrong. Machines don't have the same emotions, but nowadays are imparted and tried to be implemented in the machines so that the machines become smart enough and help humans make decisions. The emotion classification in the field of computer technology is called "Sentiment Analysis (Pansy and Rupali, 2021), opinion mining, or text mining (Bilal and Saad, 2017)," but the analysis is often combined with recommender systems. The sentiment analysis does the smart work of classifying the emotions into positive, negative, or neutral, whereas the recommender system makes smart suggestions based on some specific criteria the user is searching for. Basically, the sentiment analysis works on the principle of the NLP (Lavanya and Sasikala, 2021) for the classification of the text-based emotions given by online community users. The paper herein flows in the following sections difference between the sentiment analysis (Gang *et al.*, 2014) and recommender system, related work in the sentiment analysis discussing the research done to date, the research gap to implement the new method to perform sentiment analysis, methodology to actually implement the sentiment analysis, conclusion, and references.

Recommender System V/S Sentiment Analysis

It aims to provide users with personalized recommendations based on their past behavior and preferences. A clear working of recommender system and the sentiment analysis is described in Figure (1).

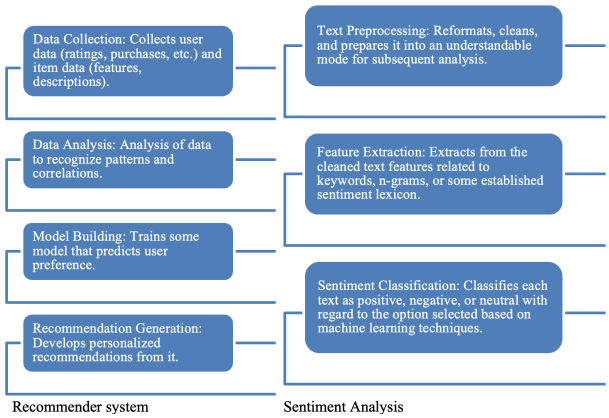

Fig. 1: Working of recommender system and sentiment analysis

Examples of the recommender system (Roy and Dutta, 2022) are Product recommendations on e-commerce websites, Movie recommendations on streaming platforms, and Friend suggestions on social media.

Whereas sentiment analysis (Kakuthota *et al.*, 2021) is an intention to classify sentiment and opinions expressed in any given text. Sentiment can be classified based on either the document level, aspect level, word level, sentence level, aspect level (Akhtar *et al.*, 2016; Zenun *et al.*, 2020), or emotion level. For example: Analysis of customer reviews to gauge product satisfaction, Brand sentiment monitoring on social media, Determining overall sentiment towards news articles, etc. Basic differences between two approaches are discussed in Table (1).

Table 1: Difference of recommender and sentiment analysis

Feature	Recommender Systems	Sentiment Analysis
Primary Goal	Personalized recommendations	Sentiment classification
Input Data	User ratings, purchase history, item metadata	Textual data (reviews, comments, social media posts)
Output	Recommended items or content	Sentiment polarity (positive, negative, neutral)
Techniques	Collaborative filtering, content-based filtering, and hybrid approaches	Natural language processing, machine learning (e.g., Naïve Bayes, SVM, deep learning)

Related Work

The authors have used the sentiment analysis technique to analyze the emotion in the underlying business emails sent and what could be the potential sentiment in response. Linear SVM and Vader sentiment analysis tools have been used for the experiment. Vader is used for getting the sentiment score whereas the analysis is performed using the Linear SVM. Basically, the analysis is done for future enhancement, customer support, improve the response given to the customer. In the future, the authors have suggested using this experiment to apply in the non-technical areas as well as to implement algorithms other than the ones used (Borg and Boldt, 2020).

The authors have used the deep learning techniques for the sentiment bifurcation of the movie reviews fetched from the IMDb platform. The study includes the sentiment classification using bidirectional long short-term memory (BiLSTM) (Murthy *et al.*, 2020) model only. The model performed exceptionally well as compare to the other traditional models (Zabit and Begonya, 2020).

The authors have used the sentiment factor for the recommendation purpose and have used this on the product reviews obtained from the Yelp dataset. The HowNet sentiment dictionary was used by the authors to calculate the sentiment of the item and have extended it for their experiment. Specific differentiation is put by the authors on the sentiment and features of the item. For the future the future the authors have suggested using deep learning methods for implementation purposes along with real-world dataset implementation (Lei *et al.*, 2016).

Aspects of any product have a distinguished identity when it comes to sentiments, as the sentiments are representative of a particular product aspect only. The authors have discussed the problems faced with the aspects of extraction and also the challenges faced. How to access aspects from the multi-word, what can be the effect of time on the aspect, factors affecting the aspect, and much more? A comprehensive study is presented in the paper, along with the problems and their solution conducted by various authors by means of the research paper (Ambreen *et al.*, 2022).

The authors have presented a systematic study of the deep learning algorithms used in sentiment analysis. No particular experiment has been performed apart from the secondary study performed from the research papers. From the study conducted, it has been found that LSTM and CNN are the maximum used algorithms. A total of 112 papers have been considered by the authors. The authors have described a complete process of the sentiment analysis, levels of the analysis, how the algorithms are useful and their advantages. In the end, the authors have formulated the research questions to be considered for the research (Alexander *et al.*, 2021).

The KnowMIS-ABSA model, introduced by D'Aniello *et al.* (2022), presents a novel perspective for Aspect-Based Sentiment Analysis (ABSA), which aims to clear confusion concerning key concepts in the area. The model stresses that separating sentiment, affect, emotion, and opinion is very essential since they are distinct concepts that would require different metrics and techniques for proper measurement (D'Aniello *et al.*, 2022). The most curious part here is that while other papers delve into ABSA techniques with various neural network architectures or attention mechanisms, the KnowMIS-ABSA model has taken a step back to re-evaluate the elementary concepts of ABSA. This would stand in contrast to the trend of using more and more complex models, such as the MHAKE-GCN model (Cui *et al.*, 2023) or the AS-Reasoner (Ning *et al.*, 2019); both of the latter directions tend to focus on improving the performance of sentiment classification tasks by means of slight modifications of neural network architectures. To summarize, the KnowMIS-ABSA model affords ABSA a new angle by presenting a reference model to insist upon the distinctness of tools, metrics, and assemblages of certain dimensions of opinion. The fresh perspective offered by KnowMIS-ABSA could help in making a number of ABSA techniques more nuanced and accurate. However, it must be noted that the paper showcases a qualitative case study instead of well-crafted

quantitative experiments, as is usual in other ABSA research.

The implementation of deep learning approaches for sentiment rating predictions for product reviews has shown a promising possibility with increases in accuracy and results over traditional machine learning methodologies. Most of the studies highlight that deep learning models worked well in sentiment analysis of product reviews. In one such study, it was shown that LSTM Networks achieved 94% accuracy on Google Play consumer reviews in Chinese, outperforming Naïve Bayes (74.12%) and Support Vector Machine (76.46%) approaches (Min-Yuh and Yue-Da, 2017). Another study on LSTM applied to customer review sentiment prediction achieved results with an accuracy of 93.66% (Krishna Kumar, 2021). A capsule-based RNN approach has attained an impressive 98.02 percent accuracy score, which is better than CNN and RNN-based models (Md Shofiqul *et al.*, 2024). Interestingly, different deep learning architectures exhibited different proficiencies across datasets. These results suggest that hybrid models integrating multiple deep learning techniques might be beneficial in producing improved performance in sentiment analysis tasks. An example of this would include the CRDC (Capsule with Deep CNN and Bi-structured RNN) model, which performs better than its counterparts across different databases: IMDB with an accuracy of 88.15%, Toxic with 98.28%, CrowdFlower with 92.34%, and ER with 95.48% (Md Shofiqul *et al.*, 2024). In summary, deep-learning approaches are showing better performance as compared to traditional machine-learning methods in sentiment analysis on product reviews. It is the basic nature of deep learning models that allows them to catch both the syntax and semantics of text without high-level feature engineering that leads to better results (Do *et al.*, 2019). However, there is potential for improvement, and consequently, research continues in better architectures and techniques for improving sentiment analysis accuracy and efficiency.

The COVID-19 pandemic has significantly changed social life to such an extent that public sentiment studies are now being carried out based on more advanced natural language processing techniques, such as BERT. BERT is merely one of several methods for performing sentiment analysis studies on COVID-related social media data. It has been used, for instance, to study 999,978 posts on Weibo between January and February 2020 through the help of an unsupervised BERT model in order to classify sentiments as positive, neutral, and negative.

This research (Mrityunjay *et al.*, 2021) identified four major aspects in which the public showed concern regarding the origin of the virus, its symptoms, production activity, and public health control. In another study, researchers collected over 3 million tweets from January 2021 to February 2022, using BERT as the basis of their sentiment analysis toward COVID-19 vaccination in the United States. According to this study, 35% of the tweets displayed a negative attitude, while the positive response accounted for 65% of that of vaccination.

Interestingly, some studies also compare BERT performance to different machine learning techniques. For instance, one Twitter sentiment analysis performed during the COVID-19 pandemic peak concluded that Bi-LSTM achieved higher accuracy (0.87) than traditional machine-learning models. Another study, though, proposed an approach using sentiment analysis and key entity detection based on BERT meant for acquiring financial text streams, which outperformed classical methodologies like SVM, LR, and NBM (Lingyun *et al.*, 2021).

Research Gap

The authors in various researches have used the basic models for the sentiment analysis but the hybrid models or the stacking models are not discussed at length till now and herein we have implemented the stacking model for the sentiment analysis.

Materials

The Jupyter Notebook environment, which offered an interactive platform for data analysis and model development, was used to conduct the research using the Python programming language. Several libraries from the Python ecosystem were used to carry out the experiments, including:

TextBlob: For fundamental sentiment polarity and subjectivity analysis; Flair: A potent natural language processing library for sophisticated sentiment classification using contextual word embeddings

Scikit-learn: For putting traditional machine learning models like Random Forest, K-Nearest Neighbors (KNN), AdaBoost, and Logistic Regression into practice

Imbalanced-learn: For managing data imbalance using methods like SMOTE (Synthetic Minority Oversampling Technique)

Pandas, NumPy, Matplotlib, and Seaborn: For preprocessing, statistical analysis, and visualization of data.

Unwrangle.com, a platform that curates and makes available real-world, structured datasets, kindly contributed the dataset used in this study. The information included consumer reviews of electronic devices, particularly televisions, that were gathered from Amazon between 2016 and 2023.

The Anaconda distribution, which conveniently bundled the necessary libraries and tools, was used for all experiments, guaranteeing a reliable and consistent execution environment.

Methodology

The Sentiment analysis can be carried out in various fields, be it e-commerce, co-operate, entertainment, information technology, etc. Here, we have worked on electronic products for the purpose of sentiment analysis. But which algorithm to use and which will be the best for the task depends on the accuracy given by it. In the case of the rating prediction, the algorithms I have used are random forest, KNN, Naïve Bayes, Logistic Regression, SVM, and Adaptive Boosting (AdaBoost). The dataset is collected from the Unwrangle data provider firm. The reviews under consideration are television reviews ranging from the year 2016 to 2023, fetched from the Amazon website. The data preprocessing phase handles the missing values by dropping them from the dataset for accurate prediction and no misleading information if included. The sentiments are divided into numerical values for the usage in the experiment. An undersampling technique like SMOTE is used for the imbalanced data, and meaningful numerical values are generated for the aspects included in the reviews using the TF-IDF.

Algorithm Selection

The dataset used in the experiment is not extremely large, for which Random Forest, KNN, and Ada Boost fit well for the implementation as compared to the XG Boost or LSTM. We have generated a binary label of the sentiment rating from the reviews provided, and Random Forest, k-NN, and Ada Boost work straight to numerical attributes such as TF-IDF vectors. They do not need the written data to inherently follow a sequential order while LSTM is designed for the sequential or contextual relationship of data. Also the models used are easier to interpret as compared to others. The models are less demanding when compared to the hardware required and can be executed on the minimum basic hardware configuration, whereas XGBoost is iterative boosting in nature, making it computationally intensive, requiring careful tuning of hyperparameters to avoid overfitting. LSTM requires a long processing time as it uses sequential data processing, especially for larger vocabularies or embedding layers.

The recent algorithm AdaBoost (Feng, 2019) stands for Adaptive Boosting, uses the learning method that combines multiple weak learners to create a strong, accurate model. Working iteratively training weak learners, focusing on the instances that the previous learners misclassified.

Working on the AdaBoost Algorithm

1. Initialization

   Assign equal weights to each training instance:

   $$w\_i = 1/N \quad (1)$$

   where N is the number of training instances.

2. Training Weak Learners:

   Train a weak learner (e.g., decision tree) on the weighted training data.

   Training Weak Learner t:

   Train a weak learner h_t on the weighted training data.

   The weak learner's predictions are used to calculate its error rate.

   Calculate the error rate of the weak learner:

   $$\varepsilon\_t = \Sigma\left(w\_i * |h\_t\left(x\_i\right) - y\_i|\right) / \Sigma\left(w\_i\right) \quad (2)$$

   Calculate the weight of the weak learner:

   $$\alpha\_t = 0.5 * ln\left(\left(1 - \varepsilon\_t\right) / \varepsilon\_t\right) \quad (3)$$

3. Updating Weights:

   Increase the weights of misclassified instances.

   Decrease the weights of correctly classified instances.

   This ensures that the next weak learner focuses more on the difficult instances.

   Update the weights of the training instances:

   $$w\_i = w\_i * exp\left(-\alpha\_t * y\_i * h\_t\left(x\_i\right)\right) \quad (4)$$

   Normalize the weights to ensure they sum to 1.

4. Combining Weak Learners:

   Each weak learner is assigned a weight based on its accuracy.

   The final prediction is made by combining the weighted predictions of all weak learners.

   The final prediction for a new instance x is given by:

   $$H\left(x\right) = sign\left(\Sigma\left(\alpha\_t * h\_t\left(x\right)\right)\right) \quad (5)$$

The AdaBoost algorithm has underperformed due to the class imbalance in our because it works by assigning higher weights to misclassified samples in each iteration. If a minority class is initially misclassified, subsequent iterations might overly focus on these minority samples, leading to overfitting or underperforming on the majority class. AdaBoost struggles to include more complex features of the data. AdaBoost may overfit the training data, especially if the number of estimators is large or if there is a mismatch between the complexity of the data and the model.

Stacking Model

Multiple models can be combined to form a single more efficient model using techniques like Bagging (Bootstrap Aggregating), where predictions are combined from multiple models after executing on the random subsets of data or boosting is correcting the errors of previous models and combining the predictions of weak learners and to get the accuracy is the only aim

of the prediction models, and this accuracy can be more accurate when more than one model is combined to form a single model. This technique is known as stacking (Sharaf and Anas, 2024). The predictions obtained are used as the features to train the blender in the new model to predict the values.

In the dataset used, the review text is a mix of text data, which requires transformation into numerical features. The review rating is a numeric label that indicates positive or negative sentiment. There is likely some class imbalance, including more positive than negative reviews. Hybrid models often involve neural networks or ensembles with heavier computational demands. Neural models like LSTM have many parameters to train, which can overfit small datasets. The hybrid model doesn't work well when the dataset provided is not very large. Hybrid models are more complex, making them harder to interpret compared to stacking. The dataset used in the study is of moderate size wherein computational efficiency is required, and in this case, the stacking model performs well. When dealing with text-based data, the TF-IDF or word embeddings can be directly used with traditional ML models in stacking, whereas the Hybrid models often require embeddings and specialized layers, increasing complexity and thus increasing the computation time, which makes the stacking Figure (2) a more obvious choice compared to the hybrid.

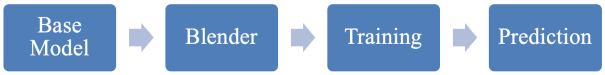

Fig. 2: Stages of the stacking model

Results and Discussion

The values obtained from the execution results clearly depicts that the Random forest has given the highest accuracy score of 88% as compared to KNN having 87% accuracy and AdaBoost having 77% accuracy which is less than the two models.

The Figure (3) shows the graphical representation of the Table (2), which represents model performance across the metrics (Precision, Recall, F1-Score) for each class (-1, 0, 1, and Overall). Each graph highlights how well Random Forest, KNN, and AdaBoost performed for each metric.

We have used Stratified K-Fold Cross-Validation (Jerzy *et al.*, 2022) to evaluate the models. As this technique ensures that each fold maintains the class distribution of the dataset, making the evaluation more consistent and reliable. The cross-validation (Jing, 2020) results of the findings are described in Table (4).

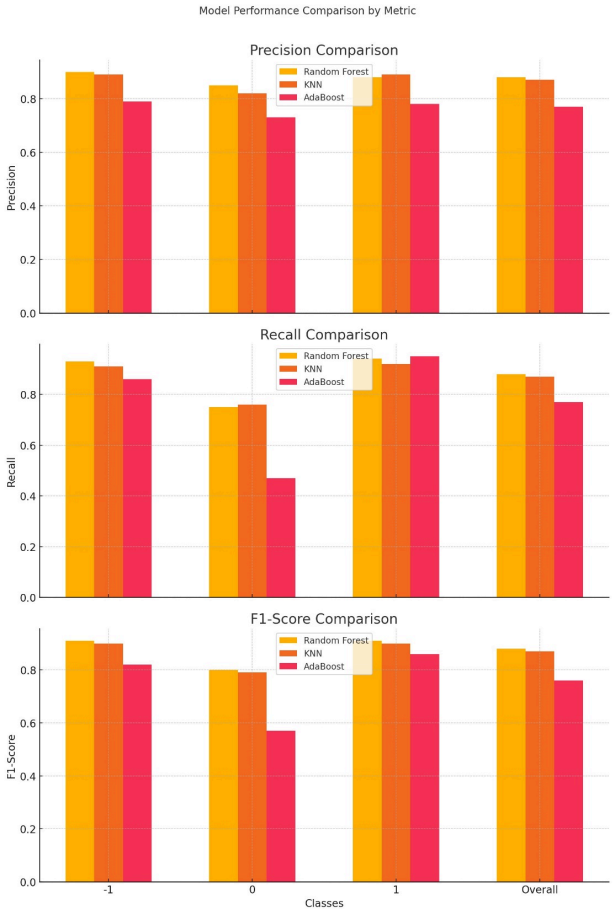

Fig. 3: Precision comparison, Recall comparison and F1 score comparison

Combining Random Forest and Logistic Regression in a stacking ensemble could leverage their strengths. AdaBoost and Logistic Regression can benefit from parameter tuning or feature engineering.

To get more accurate either of the following steps can be done Hyper parameter tuning, feature engineering, advances model selection, and feature engineering.

Here, we have implemented the hyperparameter tuning, which results in the following result

Random forest accuracy: 88%:

$$KNN\ with\ power\ size\ [Manhattan = 1, Euclidean = 2]$$
$$resulting\ in\ accuracy:\ 87 \quad (6)$$

Table 2: Results of the Random forest, KNN, and AdaBoost algorithm

Algorithm	Random Forest			KNN			AdaBoost		
Class	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score
-1	0.9	0.93	0.91	0.89	0.91	0.9	0.79	0.86	0.82
0	0.85	0.75	0.8	0.82	0.76	0.79	0.73	0.47	0.57
1	0.88	0.94	0.91	0.89	0.92	0.9	0.78	0.95	0.86
Overall	0.88	0.88	0.88	0.87	0.87	0.87	0.77	0.77	0.76

And replacing the AdaBoost algorithm with the XGBoost gradient power algorithm which performs exceptionally well with accuracy of 94%.

The traditional models were implemented and performed well for the prediction values of the emotion which were wrapped in the reviews of the customer. But even more efficient accuracy score can be obtained by combining models together so that the strength of various algorithms can be combined and an exceptional result can be obtained. This can be done using boosting, stacking and bagging.

Here we have used the stacking wherein the logistic regression, KNN and random forest models are used as the base models to train the Gradient Boosting model to get the combined accurate result.

After the implementation of the stacking model, the classification report for your stacking model shows that it performs very well for class 1, but it struggles significantly with classes -1 and 0. This suggests an imbalance in the dataset or an issue with how the model is prioritizing certain classes, and the result obtained is displayed in Tables (3-4).

Table 3: Result of the stacking model

Class	Precision	Recall	F1-Score	Support
-1	0.43	0.12	0.19	248
0	0.47	0.04	0.07	476
1	0.94	1	0.97	9674
Accuracy			0.93	10398
Macro Avg	0.61	0.39	0.41	10398
Weighted Avg	0.9	0.93	0.91	10398

Table 4: Stratified K-fold cross-validation evaluation result

Model	Accuracy	Precision (Macro)	Recall (Macro)	F1 (Macro)
Random Forest	0.8337	0.3862	0.4322	0.3976
KNN	0.4197	0.3517	0.4284	0.2498
AdaBoost	0.5556	0.3596	0.4684	0.3059
Logistic Regression	0.5679	0.3627	0.4751	0.3138

The stacking model uses the important characteristics of different models to get more accurate results, thus improving the overall performance of implementation, be it in classification or regression, as compared to the single model. The results of the experiment conducted using stacking model is included in the Table (3). Basically, the stacking model uses the base model and Metamodel for its implementation wherein the stacker (Metamodel) learns from the base model and makes the desired adjustments to the weakness to deal with the underfit or overfit issues encountered in single model implementation, alleviating these problems by blending the outputs for the best outcome. The stacking model leverages the model diversity by combining multiple models like logistic regression, deep learning (Md Shofiqul *et al.*, 2024), or random forest. The basic principle of stacking is to work with any combination of traditional Machine Learning models, neural networks, or even more precise pre-trained models like BERT (Lingyun *et al.*, 2021). Cross-validation ensures that the meta-model generalizes well to unseen data. Using the cross-validation on the training base models the stacker avoids the overfitting to the training data. The stacking model handles real-life problems very well, like the recommender system, text classification, fraud detection, and many more. The proof of the validation is included in the in Table (4) which included the result of the Cross Validation implementation. Thus, stacking can be used when there are diverse models capturing different aspects of the data, improving the performance of existing models, or there are sufficient computational resources available because stacking is computationally expensive.

Conclusion

The experimental results unequivocally show that the Random Forest algorithm outperformed both KNN (87%) and AdaBoost (77%), achieving the highest accuracy of 88% among individual models. Metrics like precision, recall, and F1-score for various classes further demonstrated Random Forest's superior and reliable performance. Ensemble strategies and hyperparameter tuning were investigated to improve model performance. By substituting XGBoost for AdaBoost, accuracy increased to 94%, demonstrating the potential of sophisticated gradient boosting methods. Additionally, stacking was implemented with Gradient Boosting as the meta-learner and Logistic Regression, KNN, and Random Forest as base learners. Class imbalance issues were indicated by this ensemble strategy's poor performance for minority classes (-1 and 0), despite its high overall accuracy of 93%.

With every factor considered, the stacking model shows how well model diversity can be used to enhance classification performance. To improve performance across all classes, future developments could concentrate on correcting data imbalance and improving model selection. The findings demonstrate the effectiveness of ensemble learning, particularly stacking, for sentiment or emotion classification tasks in customer review analysis.

Future Enhancements

Still the other algorithms can be included in the stacking for the combination model also bagging and boosting can be implement on the data for the prediction of the sentiment. R^2 gives a percentage of how much of the target variability is explained while the F1 score ensures a classifier performs well on critical categories in imbalanced datasets.

The R^2 used to find the accuracy uses which works on the formula:

$$R^2 = 1 - \left(SS\_res/SS\_tot\right) \quad (7)$$

where, *SS_res* is the sum of squared residuals, and *SS_tot* is the total sum of squares. The value hence calculated using the formula suffers the problem like adding more features always increases R^2, even if the features are not meaningful, lack generalizability, and don't work well for non-linear models unless adjusted. These problems can be improved by re-forming the formula as Adjusted R^2 Formula:

$$R^2\_adjusted = 1 - [(1 - R^2) * (n - 1)/(n - p - 1)] \quad (8)$$

where, *n* is the number of observations and p is the number of predictors.

Acknowledgment

The data used in this study was kindly provided by Mr. Ranauq Singh of Unwrangle, for which the authors are truly grateful. His assistance and input have been crucial to finishing this research. Additionally, I Asst. Prof. Jyoti Verma would like to express my sincere gratitude to Dr. Jaimin Undavia, Ph.D. Guide, for his unwavering support and direction during the research process. Special thanks is extended to fellow research scholars Assistant Professors Dhatri Raval and Mubina Malik for their invaluable advice, spirit of cooperation, and continuous support. A heartfelt thanks to my student Prem Mali who has been the bridge between the data provider and my research.

We are grateful to the publisher of the paper and the editors who have made this research a piece of art and worthy to be considered in publication in the journal.

Funding Information

The research presented in this study has not received any funding or support from any funding agency or individual. All experiments were conducted independently, using self-generated data and resources. Except that the data are provided without any profit benefit by the owner and founder of Unwrangle Mr. Ranauq Singh.

Author's Contributions

Jyoti S Verma: Participated in all experiments, coordinated the data analysis.

Jaimin N Undavia: Article or reviewing it critically for significant intellectual content.

Ethics

The content, data, and methodology utilized in this study are all free of ethical issues or conflicts, according to the authors. Every piece of data used was gathered morally and with the proper authorization. Additionally, the authors attest that no problems are expected when this study is published.

References

Akhtar, M. S., Ekbal, A., & Bhattacharyya, P. (2016). Aspect based sentiment analysis in Hindi: Resource creation and evaluation. *Proceedings of the 10th International Conference on Language Resources and Evaluation, LREC 2016*, 2703-2709. https://doi.org/10.1007/978-3-319-75487-1_19

Alexander, L., Cagatay, C., & Bedir, T. (2021). Systematic reviews in sentiment analysis: a tertiary study. *Artificial Intelligence Review*, *54*(7), 4997-5053. https://doi.org/10.1007/s10462-021-09973-3

Ambreen, N., Yuan, R., Lianwei, W., & Ling, S. (2022). Issues and Challenges of Aspect-Based Sentiment Analysis: A Comprehensive Survey. *IEEE Transactions on Affective Computing*, *13*(2), 845-863. https://doi.org/10.1109/taffc.2020.2970399

Bilal, S., & Saad, Saad. (2017). Sentiment Analysis or Opinion Mining: A Review. *International Journal on Advanced Science, Engineering and Information Technology*, *7*(5), 1660-1666. https://doi.org/10.18517/ijaseit.7.4.2137

Borg, A., & Boldt, M. (2020). Using VADER sentiment and SVM for predicting customer response sentiment. *Expert Systems with Applications*, *162*, 113746. https://doi.org/10.1016/j.eswa.2020.113746

Cui, X., Tao, W., & Cui, X. (2023). Affective-Knowledge-Enhanced Graph Convolutional Networks for Aspect-Based Sentiment Analysis with Multi-Head Attention. *Applied Sciences*, *13*(7), 4458. https://doi.org/10.3390/app13074458

D'Aniello, G., Gaeta, M., & La Rocca, I. (2022). KnowMIS-ABSA: an overview and a reference model for applications of sentiment analysis and aspect-based sentiment analysis. *Artificial Intelligence Review*, *55*(7), 5543-5574. https://doi.org/10.1007/s10462-021-10134-9

Do, H., Prasad, P., Maag, A., & Alsadoon, A. (2019). Deep Learning for Aspect-Based Sentiment Analysis: A Comparative Review. *Expert Systems with Applications*, *118*, 272-299. https://doi.org/10.1016/j.eswa.2018.10.003

Feng, X. (2019). Research of Sentiment Analysis Based on Adaboost Algorithmof Sentiment Analysis Based on Adaboost Algorithm. *2019 International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI)*, 279-282. https://doi.org/10.1109/mlbdbi48998.2019.00062

Gang, W., Sun, S., Jian, M., Kaiquan , X., & Jibao, G. (2014). Sentiment classification: The contribution of ensemble learning. *Decision Support Systems*, *57*, 77-93. https://doi.org/10.1016/j.dss.2013.08.002

Jerzy, W., Cole, G., & Thomas, M. (2022). K-fold cross-validation for complex sample surveys. *Stat*, *11*(1), 454. https://doi.org/10.1002/sta4.454

Jing, L. (2020). Cross-Validation With Confidence. *Journal of the American Statistical Association*, *115*(532), 1978-1997. https://doi.org/10.1080/01621459.2019.1672556

Kakuthota , R., Ramalingam, H. M., Pavithra, M., Advi, H. D., & Maithri, H. (2021). Sentimental analysis of Indian regional languages on social media. *Global Transitions Proceedings*, *2*(2), 414-420. https://doi.org/10.1016/j.gltp.2021.08.039

Krishna Kumar, M. (2021). Sentiment analysis for product rating using a deep learning approach. *2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS)*, 121-126. https://doi.org/10.1109/icais50930.2021.9395802

Lavanya, PM., & Sasikala, E. (2021). Deep Learning Techniques on Text Classification Using Natural Language Processing (NLP) In Social Healthcare Network: A Comprehensive Survey. *2021 3rd International Conference on Signal Processing and Communication (ICPSC)*, 603-609. https://doi.org/10.1109/icspc51351.2021.9451752

Lei, L., Xueming, Q., & Guoshuai, Z. (2016). Rating Prediction Based on Social Sentiment From Textual Reviews. *IEEE Transactions on Multimedia*, *18*(9), 1910-1921. https://doi.org/10.1109/tmm.2016.2575738

Lingyun, Z., Lin, L., Xinhao, Z., & Jianwei, Z. (2021). A BERT based Sentiment Analysis and Key Entity Detection Approach for Online Financial Texts. *2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, 1233-1238. https://doi.org/10.1109/cscwd49262.2021.9437616

Md Shofiqul, I., Muhammad Nomani, K., Ngahzaifa Ab , G., Kamal Zuhairi , Z., Nor Saradatul Akmar, Z., Md Mustafizur, R., & Mohammad Ali, M. (2024). Challenges and future in deep learning for sentiment analysis: a comprehensive review and a proposed novel hybrid approach. *Artificial Intelligence Review*, *57*(3), 62. https://doi.org/10.1007/s10462-023-10651-9

Min-Yuh, D., & Yue-Da, L. (2017). Deep Learning for Sentiment Analysis on Google Play Consumer Review. *2017 IEEE International Conference on Information Reuse and Integration (IRI)*, 382-388. https://doi.org/10.1109/iri.2017.79

Mrityunjay, S., Amit Kuma, J., & Shivam, P. (2021). Sentiment analysis on the impact of coronavirus in social life using the BERT model. *Social Network Analysis and Mining*, *11*(1), 33. https://doi.org/10.1007/s13278-021-00737-z

Murthy, G., Shanmukha Rao, A., Bhargavi, A., Mounika, Bagadi, & Mounika, B. (2020). Text based Sentiment Analysis using LSTM. *International Journal of Engineering Research And*, *9*(05), 299-303. https://doi.org/10.17577/ijertv9is050290

Ning, L., Bo , S., Zhenjiang, Z., Zhiyuan, Z., & Kun, M. (2019). Attention-based Sentiment Reasoner for aspect-based sentiment analysis. *Human-Centric Computing and Information Sciences*, *9*(1), 35. https://doi.org/10.1186/s13673-019-0196-3

Pansy , N., & Rupali, V. (2021). A review on sentiment analysis and emotion detection from text. *Social Network Analysis and Mining*, *11*(1), 81. https://doi.org/10.1007/s13278-021-00776-6

Roy, D., & Dutta, Mala. (2022). A Systematic Review and Research Perspective on Recommender Systems. *Journal of Big Data*, *9*(1), 59. https://doi.org/10.1186/s40537-022-00592-5

Sharaf J, M., & Anas W , A. (2024). A Stacking Ensemble Based on Lexicon and Machine Learning Methods for the Sentiment Analysis of Tweets. *Mathematics*, *12*(21), 3405. https://doi.org/10.3390/math12213405

Zabit, H., & Begonya, G.-Z. (2020). Sentiment Classification Using a Single-Layered BiLSTM Model. *IEEE Access*, *8*, 73992-74001. https://doi.org/10.1109/access.2020.2988550

Zenun and Imran, K., Ali Shariq, I., & Arianit, K. (2020). Weakly Supervised Framework for Aspect-Based Sentiment Analysis on Students' Reviews of MOOCs. *IEEE Access*, *8*, 106799-106810. https://doi.org/10.1109/access.2020.3000739