Exploring Sentiment Analysis on Social Media Texts

Najeeb Abdulazez Alabdulkarim
Department of Computer Science, College of Computer and Information Science, Majmaah University, 11952, Saudi Arabia
441104725@s.mu.edu.sa

Mohd Anul Haq
Department of Computer Science, College of Computer and Information Science, Majmaah University, 11952, Saudi Arabia
m.anul@mu.edu.sa

Jayadev Gyani
Department of Computer Science, College of Computer and Information Science, Majmaah University, 11952, Saudi Arabia
je.gyani@mu.edu.sa (corresponding author)

Received: 12 March 2024 | Revised: 24 March 2024 | Accepted: 2 April 2024

 DOI: https://doi.org/10.48084/etasr.7238

ABSTRACT

Sentiment analysis is a critical component in understanding customer opinions and reactions. This study explores the application of sentiment analysis using Python on the Amazon Fine Food Reviews dataset to classify customer reviews as positive or negative, enabling businesses to gain valuable insight into customer sentiments. This study used and compared the efficiency of Logistic Regression, Support Vector Machines, Random Forest, XGBoost, LSTM, and ALBERT. The comparison results showed that the LSTM and ALBERT classifiers stand out with remarkable accuracy (96%) and substantial support for positive and negative reviews. On the other hand, although the Random Forest classifier had similar accuracy (96%), it exhibited lower support for positive and negative sentiments.

Keywords-sentiment analysis; classification; LSTM; ALBERT; regression; XGBoost ; SVM

I. INTRODUCTION

Sentiment analysis is very significant in this technology-powered world where people connect through social media. It is especially useful when deciding on what to buy. The process of extracting sentiments from huge bodies of textual data that people post on Amazon is important because it helps in tracking the feelings that may exist toward products, services, or events. This analysis contributes to informed decision-making processes to help businesses gauge customer satisfaction. This study aims to comprehensively evaluate and compare the effectiveness of several distinct classifiers, including Logistic Regression, Support Vector Machines, Random Forest, XGBoost, LSTM, and ALBERT, to classify Amazon reviews based on the involved sentiment utilizing the TensorFlow and Keras machine learning frameworks. The current study contributes to the existing body of knowledge by offering a comparative analysis of six classifiers, shedding light on their applicability and performance in sentiment analysis. The comparison provides businesses, researchers, and practitioners with valuable insights on sentiment analysis in the context of social networks.

Today, sentiment analysis is one of the most valuable topics because people communicate around the world using social networks that affect their purchase decisions. Measuring mentions on the large amount of text data uploaded on web platforms is key to getting a picture of the clients' attitudes toward products, services, or events. This type of analysis can enable managers to become more objective and capable of understanding client expectations in detail. This study conducts an integrated assessment and comparison of Logistic Regression, Support Vector Machines, Random Forests, XGBoost, LSTM, and ALBERT on the Amazon Reviews dataset for sentiment classification. The former serves to broaden individuals knowledge about the sentiment analysis topic through new insights, bringing benefits to businesses, researchers, and practitioners. It denotes the substantial improvement in existing sentiment analysis techniques, which are the key components of social network studies.

Advances in Web development have resulted in a dynamic and user-oriented digital landscape with continuous user participation in the creation of content and exchange of ideas through social networks, resulting in a diverse range of perspectives. Social networks act as powerful networking outlets with lots of information, where people share content and connections. Not only do they help businesses obtain facts and figures, but these portals also become strategic tools for the development of effective strategies and decision-making [1]. Sentiment extraction can be difficult, with both local and global contexts subtly coming into play and influencing how a user feels about a product. As a result, knowing how these techniques can perform, based on the type of context, is another sphere of research of great importance in both academic and practical fields [2]. In the age of user-generated content, blogs, forums, and reviews, all come under the umbrella of opinion mining and hence become the main field of focus. The prospect of applying a wide range of machine learning techniques to unravel hidden patterns and information from the growing number of unstructured texts on social networking platforms has gradually attracted the interest of researchers [1, 3-6]. Nevertheless, this task is not independent of the ever-common problems due to the vague and content-dependent nature of social media written by diverse sources.

In [7], sentiment analysis was performed on Google Play customer reviews using RoBERTa, ALBERT, and BERT and various preprocessing and optimization techniques. In [8], an attention-based bidirectional CNN-RNN deep model was implemented for sentiment analysis and compared with other DNNs, focusing on polarity detection in document-level analysis across three tweet datasets. In [9], automatic categorization of online reviews was performed deploying SVM, Random Forest, Logistic Regression, and XGBoost models on Amazon Fine Food Reviews. These studies play a pivotal role in advancing the field of sentiment analysis by devising new techniques, each with its particular role, and following multiple evaluation methods for various models that encounter specific difficulties in sentiment classification. The bidirectional CNN-RNN deep model, CNN-LSTM, and CNN-BiLSTM were superior over conventional models and standard supervised approaches. Despite these advances, existing sentiment analysis models still face several challenges, indicating the need for further model development to enhance precision.

The research objectives of this study are:

- Introduce an automated sentiment classification system based on machine learning algorithms
- Compare the performance of Logistic Regression (LR), Support Vector Machines (SVM), Random Forest (RF), XGBoost, LSTM, and ALBERT to identify the most effective approach for sentiment analysis in the context of food reviews.
- Optimize different feature representation techniques, especially word embedding, to enhance the accuracy of sentiment classification.

II. RESEARCH METHODOLOGY

Online purchasing has witnessed rapid development in recent years. As a consequence, online reviews of products have also grown extensively. User reviews involve opinions that may be recommendations or complaints. Therefore, sentiment analysis entails determining the emotional tone behind a piece of text and has become increasingly important for businesses seeking to understand customer reactions to their products or services. This study provides a sentiment classification analysis on the Amazon Reviews dataset with different ML approaches and various hyperparameter tunings. Reviews are classified as positive or negative.

A. Data Preprocessing and Exploratory Data Analysis

The Amazon Fine Food Reviews dataset, which includes a large collection of reviews from October 1999 to October 2012, was utilized [10]. The dataset contains information on users, products, and scores, with a total of 568,454 reviews. Table I provides a review sample.

TABLE I. SUMMARY OF SAMPLE REVIEWS

Id	ProductId	UserId	ProfileName	HN	HD	Score	Time	Summary
195374	B000LKVQXY	A157VRX0UUN1WU	Jim "Jim"	8	9	5	1182384000	Tastiest and best nutrition of any energy bar
559893	B002L2PGH2	A3DL36K8YVG8ZD	Sharpshooter	4	5	2	1318809600	Price BS
349938	B0006J32A0	A1UQDQQH7E2J77	Sandra K. Isbell	0	0	5	1210809600	best chewy for your best friend
560074	B000F4D50I	ACAIEIV03NBHY	J "Mom of twins"	2	2	4	1180051200	Native forest artichoke hearts
480	B000G6RYNE	A1RRHET1QIP1YK	Daniel Hill	0	0	5	1215302400	Great chip!
240934	B000NBQUNW	A32TLFBFRW3YB4	Chipmon	37	49	5	1179014400	Effective at lowering cholesterol

Integrating linguistic preprocessing and NLP techniques with Exploratory Data Analysis (EDA) is significant in sentiment analysis. Figure 1 provides a concise overview of the proposed preprocessing steps, offering a systematic approach and ensuring that the selected dataset is appropriately prepared for the intended sentiment analysis. EDA provides a deeper understanding of the dataset and helps in making informed decisions for data preprocessing purposes. EDA is seamlessly integrated into NLP to ensure the overall effectiveness of the analysis process. It is an integrated approach that guarantees a more robust and insightful analysis of textual data for sentiment classification.

Analyzing the distribution of product ratings is essential for understanding user satisfaction levels. Distribution analysis of product ratings was performed using Plotly, a powerful Python visualization library that helps create interactive and informative plots. This analysis showed that most customer ratings are positive. The particular observation sets the stage for further sentiment analysis. The initial step of the proposed preprocessing involves reading the dataset into a Pandas data frame. Pandas is a widely employed Python library for data

manipulation and analysis and an ideal tool for handling diverse datasets. After loading the data into the Pandas DataFrame, preprocessing and in-depth analysis were carried out. This step sets the foundation for subsequent stages in the data preparation process in sentiment analysis. The Summary, Text, and Score columns provide crucial information for sentiment analysis.

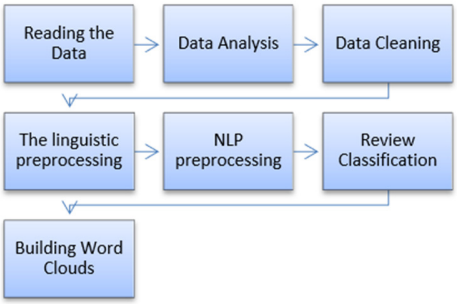

Fig. 1. Preprocessing and EDA steps.

Cleaning data is a crucial step in the preprocessing phase, encompassing various tasks, such as handling missing data, detecting outliers, addressing data imbalance, removing duplicates, reducing noise (irrelevant characters, symbols, or formatting issues), and standardizing text (lowercasing and abbreviation handling). Linguistic preprocessing plays a significant role, akin to feature selection, as it eliminates irrelevant elements from the text data. The linguistic preprocessing strategies applied to refine the reviews include:

- URLs: Since URLs do not carry sentiment, removing them helps reduce noise in the dataset.
- Stop Words: Eliminating words that do not contribute much to the classification process, such as articles ("a," "an," and "the") and prepositions.
- Numbers: Numbers typically do not have a significant impact on sentiment analysis. Their removal improves efficiency and reduces noise.
- Other Users' Mentions: Mentions of other users often have no bearing on the polarity of the review and can be ignored.
- Hashtags: In general, hashtags do not influence the classification process and were removed to streamline the analysis.

Embedding EDA through the creation of word clouds represents a ground for a valuable technique for initial exploratory analysis, offering insights into the key themes present in reviews. However, with proper preprocessing, distortions like stop words and punctuation can be removed. Word clouds were generated using a smaller (10%) sample of the data. These word clouds reveal prevalent terms, such as "taste," "flavor," "product," "good," and "coffee". Two separate clouds were constructed for positive and negative reviews, along with the basic terms associated with each sentiment. This is a way to obtain a qualitative understanding of the language utilized in different sentiments. It is important to recognize that sentiment analysis is an iterative process. The preprocessing steps portrayed in Figure 1, which resulted in the construction of word clouds, were followed. However, these steps are not strictly unidirectional. The nature of text data may require revisiting earlier steps, such as data analysis and cleaning, if new patterns and challenges emerge during review classification. This is mainly a cyclical and iterative process that combines analysis and refinement for a comprehensive treatment of text data.

NLP involves the creation of models that enable computers to comprehend, interpret, and generate human language. This capability is often employed in the monitoring of social media platforms to identify the sentiment or emotional tone of the text. The integration of linguistic preprocessing and NLP techniques is instrumental in improving classification accuracy. This study incorporated the following NLP techniques:

- Tokenization: In this phase, the reviews were segmented into tokens or terms by removing commas, symbols, white spaces, etc. It is an essential process for Part-of-Speech (PoS) tags and extracting word lemmas.
- Lemmatization: The derivationally related or inflectional forms of words are reduced to their basic forms. For instance, "the boy's cars are different colors" can be lemmatized as "the boy's car be a different color." The inputs consist of a word and its PoS tag, whereas the output is the "lemma" of the word. Individual words may have different lemmas, determined by specifying the PoS tag. For example, "saw" is lemmatized as "see" if it has a noun PoS tag and as "saw" if it has a past-tense verb tag.

In the sentiment classification step, the reviews are categorized into positive and negative sentiments. Reviews with a score greater than 3 are labeled positive (+1), those with a score less than 3 are labeled negative (-1), and reviews with a score of 3 are excluded for neutrality. This process improves the understanding of sentiment in the dataset, providing a foundation for subsequent analysis.

B. Building Sentiment Classification Models

TensorFlow is an open-source ML framework developed by Google. Keras was used as an API running on top of TensorFlow. Both offer excellent support in this area. Without dependencies, the analysis was applied in parallel frameworks, simply because the classification in a particular text is separate from the other texts [11].

1) Word Embeddings

This is an approach to demonstrate words in a continuous vector space. Word-to-Vector (Word2Vc), which converts text strings into a vector of numerical values and calculates words between distances to eventually collect similar words according to their meanings, was deployed [4]. Word2Vc helps to represent texts as numerical features, representing a feature extraction. The aim is to explore several machine-learning models for sentiment classification.

2) Logistic Regression (LR)

LR is a statistical method capable of solving binary classification problems by producing the probability that an instance fits into a certain class. It is suitable for binary

classification tasks, making it a candidate model for sentiment analysis to categorize expressions as positive or negative. The model depends on the sigmoid function to calculate probabilities. In LR, each input observation is represented through several features [12]. The feature space represents the set of variables used to predict the probability of a binary outcome. Within this space, variables such as lexicon and word counts denoted as x1, x2, x3, ..., x6, play crucial roles. Lexicon refers to a collection of words or terms with associated sentiment scores that capture semantic information within text data. Meanwhile, word counts provide quantitative measures of the frequency of specific words or terms present in the dataset. These variables, when incorporated into the LR model, contribute to the estimation of the probability of a particular outcome, allowing for effective prediction and analysis in various applications, entailing sentiment analysis, text classification, and more. The dataset needs to be labeled and each review has a sentiment label (positive or negative). An LR model is trained on the data, and its performance is evaluated using metrics, such as accuracy, precision, and recall.

3) Support Vector Machine (SVM)

SVM, as a supervised learning method, is employed for both classification and regression analyses. It is also capable of classifying nonlinear data by increasing the complexity of the classifier's bit and changing the kernel value. The SVM has an *n*-dimensional space. It separates instances by a hyperplane and can find a hyperplane or a group of hyperplanes. In supervised learning, SVM generates an optimal straight line that separates between categories. It varies the orientation and position of the hyperplane to categorize the points in the space by the highest possible margin, which in turn reduces the generalization error [5]. SVM is an embedded function in numerous tools. For example, the Sequential Minimal Optimization (SMO) algorithm is an implementation of the WEKA open-source software. SVM represents one of the most powerful statistical learning techniques and is a worthy candidate algorithm for managing complex data, including high dimensions such as images and texts. An SVM model was trained and evaluated for its effectiveness in sentiment analysis [5].

4) Random Forest (RF)

RF is known for its simplicity and is applied for both regression and classification tasks. The RF classifier is made up of several random decision trees and integrates them to produce a more accurate and stable prediction model [13]. Average results are obtained from each subtree model by random sampling with replacement of the training data. The submodels are run on an independent basis without any interdependency or dependency relations. RF also differs in how it is built because it uses a different subset of the data to construct each tree. In normal decision trees, each node splits into two branches based on the optimal separation between all independent variables, providing minimum information loss from the parent tree's dataset. Split points in each node of an RF are selected from a certain fraction of the best-split point across predictors. Thus, random forests prevent overfitting, which would be usual for a deep decision tree [14]. An RF classifier was utilized and then evaluated for its performance in sentiment classification.

5) XGBoost

XGBoost, as an ensemble learning algorithm, employs an optimized version of the Gradient Boosting Decision Tree (GBDT) framework to obtain optimal predictive accuracy. Based on an ensemble of successive decision trees, each new tree corrects the error of the previous one to reduce the residual, the residual from tree-1 fed to tree-2, and so on [14]. The advantages of XGBoost include scalability, capability, and efficiency to handle diverse datasets and problems [15]. Unlike RF, in XGBoost, each tree model minimizes the residual from its previous tree model. The XGBoost tool supports customized cost functions and runs the second-order Taylor expansion of the cost function, deploying the first and second derivatives. Traditional GBDT only employs the information of the first derivative of the error. The XGBoost algorithm was applied along with fine-tuning its hyperparameters through grid search to evaluate its performance on sentiment classification.

6) ALBERT

This model has multiple layers, and its hyperparameters are fine-tuned to achieve optimal performance. ALBERT (A Lite BERT) is a deep learning model belonging to the family of transformer-based models. ALBERT is specifically designed to address efficiency concerns and reduce computational requirements compared to Bidirectional Encoder Representations from Transformers (BERT), and it is another widely used deep learning model in NLP tasks. ALBERT is a favorable choice for certain applications because it has fewer parameters while maintaining or even improving performance [16]. The involvement of ALBERT emphasizes the ongoing efforts in the field to enhance efficiency and reduce computational overhead in complex NLP tasks. The model architecture and hyperparameter tuning contribute to its effectiveness in sentiment analysis, depending on the given dataset and the particular requirements [7]. The ALBERT design executes not only word embedding, but also placement and sector installation. The installation types are as follows:

- Word Embedding: This procedure includes inscribing the info connected with each word in the input series. Word embedding captures the semantic significance of words plus their contextual connections within the provided context.
- Placement Embedding: Position installation maps the placement of each word in the series to a vector of low-dimensional thickness. This enables the model to think about the spatial setup of words along with comprehending their placements within the general series.
- Sector Embedding: Segment installation is deployed to identify whether the presently inscribed word comes from the same sentence as the coming before words. This division helps to recognize the framework together with partnerships between various sectors within the input information.

The ALBERT design distinguishes itself by having a smaller-sized installing layer, in contrast to the BERT design's installing layer criterion. This decrease in dimension is accomplished by eliminating repetitive criteria that do not dramatically increase its efficiency. This optimization improves

its performance and reduces computational needs [16]. The structured installation layer is a vital attribute of ALBERT, ameliorating its total effectiveness in numerous NLP tasks.

7) Long-Short-Term Memory (LSTM)

LSTMs are designed to capture long-term dependencies in a sequence of words using three different gate types. This architecture allows the model to process input sequences of variable lengths accommodating sentences with different word counts. LSTMs can automatically learn relevant features without the need for manual feature engineering, enhancing their adaptability and performance [2, 8]. Unlike traditional RNNs, where neurons are connected in a directed cycle, LSTMs exhibit a chain-like structure. The three gate types of LSTM play a critical part in preserving and regulating information within each node state, enabling it to capture complex dependencies in sequential data. This structural improvement over traditional RNNs increases its ability to handle long-range dependencies and mitigates the disappearing gradient problem. The training process involves specifying hyperparameters, such as the number of hidden units, the number of layers, and the learning rate. These hyperparameters are tuned to optimize the performance of the LSTM model, thus ensuring effective learning and accurate sentiment classification for the dataset. The use of LSTM in sentiment analysis highlights its competence in handling sequential data and extracting meaningful features for classification tasks.

C. Hyperparameter Tuning

For each model, hyperparameter tuning was performed using grid search to enhance model accuracy and effectiveness. Scikit-learn provides the GridSearchCV class, allowing a comprehensive search over a specified parameter grid. The performance of classifiers can be significantly improved by optimizing parameters [17]. At first, the models were analyzed employing default hyperparameters. Then, they were tested deploying hyperparameter tuning algorithms.

III. RESULTS AND DISCUSSION

Initially, a thorough preprocessing was carried out to clean and prepare the data for analysis. Additionally, an EDA was performed to gain insight into the distribution and characteristics of the dataset. With a well-prepared dataset, six distinct classifiers were built and evaluated. The classifiers were designed to differentiate customer opinions, categorizing them as positive or negative. Each classifier underwent a rigorous training and testing process to ensure robust performance. A comprehensive analysis of performance metrics is provided, including accuracy, precision, recall, and F1-score, to provide deep awareness of the strengths and limitations of each classifier.

A. Exploratory Data Analysis Results

Figure 2 displays the temporal distribution of the reviews. This distribution reflects the inherent trends, patterns, or potential seasonality of the data. The graph porttrays a notable increase in the volume of reviews over the years. The upsurge trend started in 2007 and showed consistent and substantial growth, which reached its peak in 2013. Figure 3 demonstrates how the review scores are distributed within the dataset. A scale of 1 to 5 is typically used in Amazon reviews. The overall sentiment polarity of the reviews discloses a skewed distribution toward higher scores, especially 5, suggesting a generally positive sentiment. Figure 4 offers an overview of the data quality, illustrating the number of missing values per column in the dataset. Missing values influence the quality of the analysis, and it is useful to identify any features that may have a high number of missing data. This distribution allows for making informed decisions about handling missing data while pre-processing.

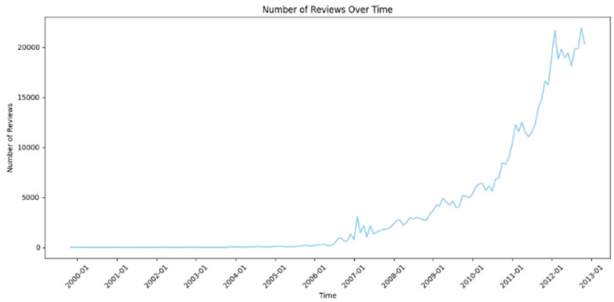

Fig. 2. Number of reviews over time.

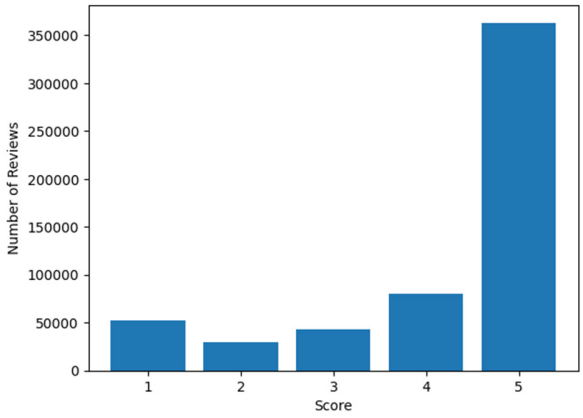

Fig. 3. Distribution of scores.

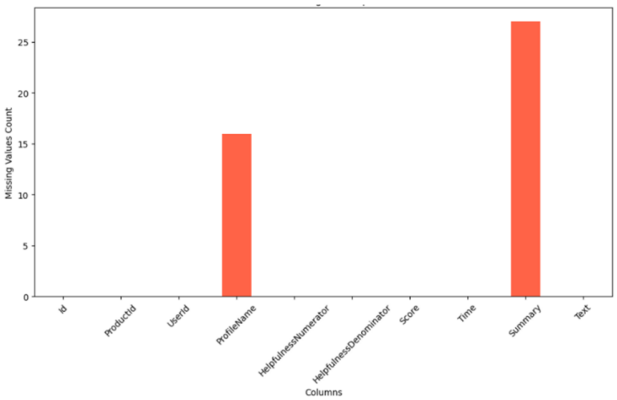

Fig. 4. Number of missing values per column.

Figure 5 exhibits the common bigrams in the dataset. Analyzing bigrams is often part of NLP tasks, including sentiment analysis. Bigrams refer to pairs of consecutive words

that occur together in a text. The grouping involves two adjacent words. Figure 6 manifests a correlation matrix between words. Figure 7 shows the distribution of letter count among the reviews. Understanding the distribution of letters helps analyze the length of reviews, allowing for the identification of outliers or patterns. This helps to decide how to handle text data during preprocessing (e.g., adjusting the maximum sequence lengths for model input). Longer reviews may indicate a higher level of user engagement. This distribution helps to understand the typical length of reviews on the platform. In general, it is valuable for companies to encourage users to leave more detailed feedback.

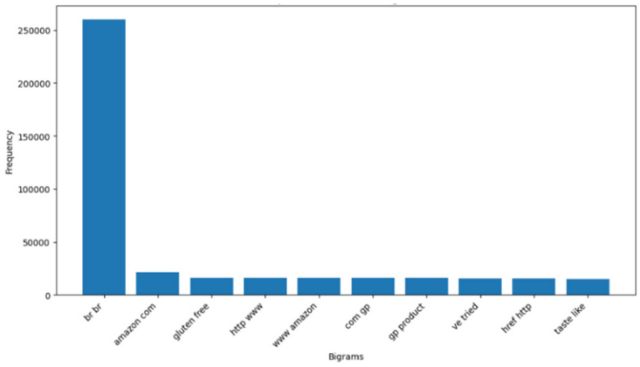

Fig. 5. Common bigrams.

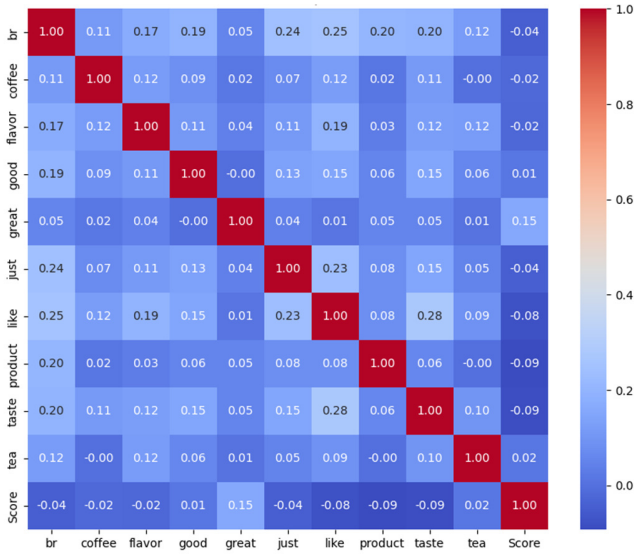

Fig. 6. Correlation between words.

The next sections detail the evaluation results of various classifiers in terms of different evaluation measures. Precision is the ratio of correctly predicted positive reviews to total positives. The recall metric represents the correctly predicted positive reviews of all reviews in the actual class. The F1-score is the weighted average of precision and recall and provides a balance between them. Accuracy represents the ratio of correctly predicted observations to their total. It also includes the support (number of instances) for the negative and positive classes to show the distribution of the dataset.

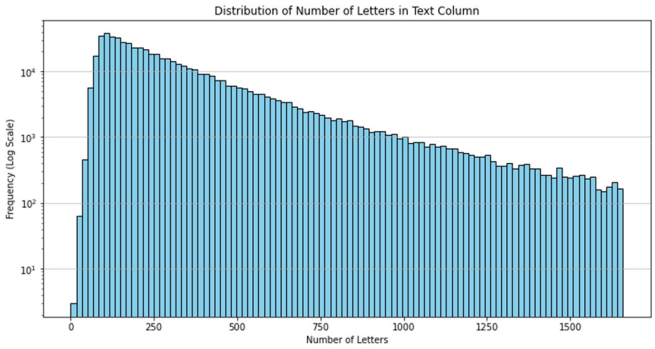

Fig. 7. Distribution of number of letters in text.

B. *Logistic Regression (LR)*

Table II provides a comprehensive evaluation of LR on the dataset. Overall, the model seems to perform well, especially in predicting positive sentiment. In terms of precision, the negative class (-1) recorded 83%, whereas the positive class (1) recorded 94%. In terms of recall (sensitivity), 67% of the actual negative and 97% of the actual positive reviews were correctly identified. The F1-score was 96% and 74% for the positive and negative classes, respectively. Support refers to the actual occurrences of the class in the dataset. According to the LR classifier, the dataset contained 16233 negative and 88925 positive reviews. The overall accuracy of the model was 93%, as the model correctly predicted positive or negative sentiment in 93% of the cases. The macro average takes the average of the precision, recall, and F1-score of both classes regardless of any class imbalance. However, the weighted average metric considers class imbalance, giving more weight to the class with more instances. Overall, the model seems to perform well, especially in predicting positive sentiments.

TABLE II. LOGISTIC REGRESSION (LR) RESULTS

	Precision	**Recall**	**Fl -score**	**Support**
-1	0.83	0.67	0.74	16233
1	0.94	0.97	0.96	88925
Accuracy	0.93	0.93	0.93	0.93
Macro avg	0.89	0.82	0.85	105158
Weighted avg	0.92	0.93	0.92	105158

C. *Support Vector Machine (SVM) Results*

Table III depicts the results of the SVM classifier. In terms of precision, about 92% of the predicted negative reviews were negative, and about 96% of the predicted positive reviews were positive. In terms of recall (sensitivity), approximately 77% of the actual negative and 99% of the actual positive reviews were correctly identified. The F1-score was 84% for negative and 97% for positive. According to the support measure, the SVM classifier identified 16233 negative and 88925 positive instances. The overall accuracy of the model was 95%. In summary, for both classes, the SVM classifier demonstrated a good overall performance with high precision and recall.

TABLE III. SUPPORT VECTOR MACHINE RESULTS

	Precision	Recall	Fl-score	Support
-1	0.92	0.77	0.84	16233
1	0.96	0.99	0.97	88925
Accuracy	0.95	0.95	0.95	0.95
Macro avg	0.94	0.88	0.91	105158
Weighted avg	0.95	0.95	0.95	105158

D. Random Forest

According to Table IV, approximately 89% of the predicted negative instances were negative and approximately 97% of the predicted positive instances were positive. In terms of recall, 82% of real negative reviews and 98% of actual positive reviews were correctly identified. The F1-score was 85% for the negative class and 97% for the positive class. The support measure indicates that 16233 instances were identified as negative and 88925 as positive. The overall accuracy of the model was 96%. The RF classifier demonstrated a good overall performance with relatively high precision, recall, and accuracy for both classes. The model seems particularly effective in predicting instances of positive sentiments.

TABLE IV. RANDOM FOREST RESULTS

	Precision	Recall	F1-score	Support
-1	0.89	0.82	0.85	16233
1	0.97	0.98	0.97	88925
Accuracy	0.96	0.96	0.96	0.96
Macro avg	0.93	0.90	0.91	105158
Weighted avg	0.95	0.96	0.96	105158

E. XGBoost Results

Table V displays the results of the XGBoost classifier. In terms of precision, about 53% of the predicted negative reviews were negative, while 99% of the predicted positive reviews were positive. In terms of recall, about 87% of the actual negative and 92% of the positive reviews were correctly identified. The F1-score was approximately 66% for negative and 95% for positive sentiments. According to the support measure, the XGBoost classifier identified 9645 negative and 95513 positive instances. The overall accuracy of the model was 92%. The XGBoost classifier exhibits a good overall performance with high precision and recall for both classes.

TABLE V. XGBOOST RESULTS

	Precision	Recall	Fl-score	Support
-1	0.53	0.87	0.66	9645
1	0.99	0.92	0.95	95513
Accuracy	0.92	0.92	0.92	0.92
Macro avg	0.76	0.89	0.81	105158
Weighted avg	0.94	0.92	0.93	105158

F. ALBERT Results

Table VI illustrates the results of the ALBERT classifier. In terms of precision, about 86% of predicted negative reviews were negative and 98% of the predicted positive reviews were positive. Regarding recall, about 87% of the real negative and 97% of the real positive reviews were correctly identified. The F1-score for the negative sentiment was 87% and for the positive was 98%. According to the support metric, the ALBERT classifier identified 20509 negative and 110945 positive reviews. Its overall accuracy was 96%. In summary, for both classes, the ALBERT classifier demonstrates high overall performance with high precision and recall.

TABLE VI. ALBERT RESULTS

	Precision	Recall	Fl-score	Support
-1	0.86	0.87	0.87	20509
1	0.98	0.97	0.98	110945
Accuracy	0.96	0.96	0.96	0.96
Macro avg	0.92	0.92	0.92	131454
Weighted avg	0.96	0.96	0.96	131454

G. LSTM Results

Table VII presents the results of the LSTM classifier. In terms of precision, 87% of the predicted negative reviews were negative and 98% of the predicted positive reviews were positive. Regarding recall, 88% of the real negative reviews and 98% of the real positive reviews were correctly identified. The F1-score was 87% for negative sentiment and 98% for positive. According to the support measure, the LSTM classifier identified 20509 negative and 110945 positive instances. The overall accuracy of the model was 96%. In summary, for both classes, the LSTM classifier demonstrated a high overall performance with high precision and recall. The support values reflect how the model is capable of identifying and classifying the reviews.

TABLE VII. LSTM RESULTS

	Precision	Recall	Fl-score	Support
-1	0.87	0.88	0.87	20509
1	0.98	0.98	0.98	110945
Accuracy	0.96	0.96	0.96	0.96
Macro avg	0.92	0.93	0.92	131454
Weighted avg	0.96	0.96	0.96	131454

H. Overall Comparison

Table VIII provides an overview of the performance of the classifiers in terms of negative class support (number of instances), positive class support, and overall accuracy. RF, ALBERT, and LSTM classifiers stand out with an accuracy of 96%. Claiming that the higher the support for the positive class is, the better the classifier generally performs, the ALBERT and LSTM classifiers outperformed the others in terms of support, as they showed the highest support for the positive (110,945) and the negative (20509) classes. These values indicate that larger amounts of instances were correctly classified as positive and negative reviews. In contrast, the XGboost classifier had the lowest support for the positive (95513) and negative (9645) classes. These results reveal their relatively lower performance in correctly classifying positive and negative reviews compared to the other classifiers.

I. Computational Speed

This study employed Google's Tensor Processing Units (TPUs) v2-8, designed to accelerate AI model training. With eight cores and 64 GiB of memory, these TPUs significantly

expedited the training process. On average, each model took less than 60 seconds to execute on the dataset.

TABLE VIII. PERFORMANCE COMPARISON OF DIFFERENT CLASSIFIERS

	-1 Support	+1 Support	Accuracy %
LR	16233	88925	93
SVM	16233	88925	95
RF	16233	88925	96
XGBoost	9645	95513	92
ALBERT	20509	110945	96
LSTM	20509	110945	96

J. Strengths and Limitations of the Study

The strengths of this study include proper data preprocessing, hyperparameter tuning of the models, high accuracy across classifiers, particularly notable for ALBERT and LSTM, along with insightful support metrics providing a detailed understanding of class distribution. However, limitations entail varying support values between classifiers, particularly lower support for XGBoost, indicating that there is a potential room for improvement in classification performance.

IV. CONCLUSION

This study sheds light on the feasibility of different algorithms in classifying and hence understanding customer opinions and reviews. The results revealed the effectiveness of every classifier, each having its own strengths and considerations. The LSTM classifier stands out with remarkable accuracy and substantial support for positive and negative reviews. Although the RF classifier achieved a similar accuracy of 96%, it exhibited lower support for positive and negative sentiments, calling for further improvement. The XGBoost classifier recorded the lowest accuracy and support of 92% among the classifiers examined. This comparison provides valuable insights into the applicability of these classifiers in sentiment analysis and an invaluable contribution to future sentiment analysis studies. However, it is important to select models that align with the specific objectives and dataset characteristics.

V. FUTURE SCOPE

Future research could explore the integration of cutting-edge models, such as GPT or other advanced NLP models, to enhance the accuracy of sentiment analysis classifiers in social networks. Additionally, investigating hybrid approaches and fine-tuning strategies could further refine these models for improved performance. Exploring domain-specific sentiment analysis tailored to different industry language patterns could provide valuable insights [18, 19]. To evaluate the effectiveness of these advancements, it would be beneficial to utilize datasets with longer texts, allowing for more comprehensive testing and validation.

ACKNOWLEDGMENT

The authors would like to thank the Deanship of Postgraduate Studies and Scientific Research at Majmaah University for supporting this work under Project No. PGR-2024-1043.

REFERENCES

[1] H. M. Chen, P. C. Franks, and L. Evans, "Exploring Government Uses of Social Media through Twitter Sentiment Analysis," *Journal of Digital Information Management*, vol. 14, no. 5, Oct. 2016, Art. no. 290, https://doi.org/10.6025/jdim/2016/14/5/290-301.

[2] L. C. Chen, C. M. Lee, and M. Y. Chen, "Exploration of social media for sentiment analysis using deep learning," *Soft Computing*, vol. 24, no. 11, pp. 8187–8197, Jun. 2020, https://doi.org/10.1007/s00500-019-04402-8.

[3] M. H. Abd El-Jawad, R. Hodhod, and Y. M. K. Omar, "Sentiment Analysis of Social Media Networks Using Machine Learning," in *2018 14th International Computer Engineering Conference (ICENCO)*, Cairo, Egypt, Dec. 2018, pp. 174–176, https://doi.org/10.1109/ICENCO.2018.8636124.

[4] A. U. Rehman, A. K. Malik, B. Raza, and W. Ali, "A Hybrid CNN-LSTM Model for Improving Accuracy of Movie Reviews Sentiment Analysis," *Multimedia Tools and Applications*, vol. 78, no. 18, pp. 26597–26613, Sep. 2019, https://doi.org/10.1007/s11042-019-07788-7.

[5] S. M. Yimam, H. M. Alemayehu, A. Ayele, and C. Biemann, "Exploring Amharic Sentiment Analysis from Social Media Texts: Building Annotation Tools and Classification Models," in *Proceedings of the 28th International Conference on Computational Linguistics*, Barcelona, Spain, Sep. 2020, pp. 1048–1060, https://doi.org/10.18653/v1/2020.coling-main.91.

[6] M. Arbane, R. Benlamri, Y. Brik, and A. D. Alahmar, "Social media-based COVID-19 sentiment classification model using Bi-LSTM," *Expert Systems with Applications*, vol. 212, Feb. 2023, Art. no. 118710, https://doi.org/10.1016/j.eswa.2022.118710.

[7] R. Sanjana, C. Tandon, P. J. Bongale, T. M. Arpita, H. Palivela, and C. R. Nirmala, "Comparative Analysis of Various Language Models on Sentiment Analysis for Retail," in *Soft Computing for Problem Solving*, Singapore, 2021, pp. 725–739, https://doi.org/10.1007/978-981-16-2709-5_55.

[8] M. E. Basiri, S. Nemati, M. Abdar, E. Cambria, and U. R. Acharya, "ABCDM: An Attention-based Bidirectional CNN-RNN Deep Model for sentiment analysis," *Future Generation Computer Systems*, vol. 115, pp. 279–294, Feb. 2021, https://doi.org/10.1016/j.future.2020.08.005.

[9] R. Singh, A. Kumar, and M. Ray, "Performances of Machine Learning Models and Featurization Techniques on Amazon Fine Food Reviews," in *Optimization Techniques in Engineering*, John Wiley & Sons, Ltd, 2023, pp. 187–199.

[10] Stanford Network Analysis Project, "Amazon Fine Food Reviews." [Online]. Available: https://www.kaggle.com/datasets/snap/amazon-fine-food-reviews.

[11] M. Khader, A. Awajan, and G. Al-Naymat, "The Effects of Natural Language Processing on Big Data Analysis: Sentiment Analysis Case Study," in *2018 International Arab Conference on Information Technology (ACIT)*, Werdanye, Lebanon, Nov. 2018, https://doi.org/10.1109/ACIT.2018.8672697.

[12] S. Halder, "Tokenization, Stemming and Lemmatization | TechGenizer," Mar. 16, 2021. https://techgenizer.netlify.app/blog/2021/03/16/tokenization-stemming-lemmatization/.

[13] D. G. Kleinbaum and M. Klein, *Logistic Regression*. New York, NY, USA: Springer, 2010.

[14] M. Al-Akhras, M. Alawairdhi, A. Alawairdhi, and S. Atawneh, "Using Machine Learning To Build A Classification Model For Iot Networks To Detect Attack Signatures," *International Journal of Computer Networks and Communications*, vol. 12, no. 6, pp. 99–116, Nov. 2020, https://doi.org/10.5121/ijcnc.2020.12607.

[15] W. Wang, G. Chakraborty, and B. Chakraborty, "Predicting the Risk of Chronic Kidney Disease (CKD) Using Machine Learning Algorithm," *Applied Sciences*, vol. 11, no. 1, Jan. 2021, Art. no. 202, https://doi.org/10.3390/app11010202.

[16] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, CA, USA, Aug. 2016, pp. 785–794, https://doi.org/10.1145/2939672.2939785.

[17] J. Li, B. Wang, and H. Ding, "Lijunyi at SemEval-2020 Task 4: An ALBERT Model Based Maximum Ensemble with Different Training

Sizes and Depths for Commonsense Validation and Explanation," in *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, Barcelona, Spain, Sep. 2020, pp. 556–561, https://doi.org/10.18653/v1/2020.semeval-1.69.

[18] E. Elgeldawi, A. Sayed, A. R. Galal, and A. M. Zaki, "Hyperparameter Tuning for Machine Learning Algorithms Used for Arabic Sentiment Analysis," *Informatics*, vol. 8, no. 4, Dec. 2021, Art. no. 79, https://doi.org/10.3390/informatics8040079.

[19] B. Ahmed, G. Ali, A. Hussain, A. Baseer, and J. Ahmed, "Analysis of Text Feature Extractors using Deep Learning on Fake News," *Engineering, Technology & Applied Science Research*, vol. 11, no. 2, pp. 7001–7005, Apr. 2021, https://doi.org/10.48084/etasr.4069.