iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 Issue: V Month of publication: May 2025

DOI: https://doi.org/10.22214/ijraset.2025.70377

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue V May 2025- Available at www.ijraset.com

Sentiment Polarity Prediction for Amazon Product Reviews Using Machine Learning and Deep Learning

Nakka Puneeth[1], Mr. M. Navalan[2]
CSE (Data Science), Sri Venkateswara College of Engineering & Technology (SVCET), Chittoor, Andhra Pradesh, India
[1]PG Researcher, Sri Venkateswara College Of Engineering & Technology (SVCET), Chittoor, Andhra Pradesh, India
[2]Assistant Professor, Sri Venkateswara College Of Engineering & Technology (SVCET), Chittoor, Andhra Pradesh, India

Abstract: The exponential growth of e-commerce platforms, particularly Amazon, has resulted in a massive volume of customer-generated reviews, making manual sentiment analysis both time-consuming and inefficient. This study proposes an automated system to predict the sentiment polarity (positive or negative) of Amazon product reviews using a combination of machine learning and deep learning techniques. The reviews are processed and converted into vectorized representations using methods such as Bag of Words (Bow), TF-IDF, Word2Vec, and their respective unigrams, bigram and weighted variants. A variety of classifiers—including Naive Bayes, Logistic Regression, Support Vector Machines (SVM), Random Forest, and Long Short-Term Memory (LSTM) networks—are trained and evaluated. Experimental results indicate that Logistic Regression with bigram TF-IDF achieves the highest test AUC of 0.9615, demonstrating superior generalization. While LSTM models show potential in capturing sequential dependencies, they require further optimization. The study emphasizes the importance of automated sentiment classification and outlines future directions such as multilingual analysis and real-time deployment using cloud platforms.

Keywords: Natural Language Processing (NLP), Support Vector Machines (SVM), Long* short *Term Memory (LSTM), Sentiment Analysis (SA), Bag of Words(Bow), Recurrent Neural Network (RNN).

I. INTRODUCTION

In the modern digital age, online product reviews play a crucial role in shaping consumer behaviour and influencing purchasing decisions. With the exponential growth of e-commerce platforms, the ability to accurately analyse and interpret customer feedback has become essential for businesses aiming to enhance product quality, improve customer satisfaction, and remain competitive in the market. This research focuses on leveraging Natural Language Processing (NLP) and sentiment analysis techniques to predict the sentiment polarity—positive, negative of Amazon product reviews. The study utilizes both structured data (such as review ratings and metadata) and unstructured data (such as textual content of reviews) to derive meaningful insights.

To build robust and accurate sentiment polarity prediction models, we employ a combination of traditional supervised machine learning algorithms and advanced deep learning techniques. The machine learning models used include Naive Bayes, Logistic Regression, Support Vector Machine (SVM), and Random Forest. These models are trained using a set of engineered textual features such as word frequencies, TF-IDF scores, and n-grams. Each algorithm offers unique advantages in terms of interpretability, computational efficiency, and performance across varying types of data distributions.

In addition to these classical approaches, we also incorporate a deep learning model—Long Short-Term Memory (LSTM), a type of Recurrent Neural Network (RNN)—which is particularly well-suited for handling sequential and context-dependent information present in textual data. LSTM is capable of capturing long-range dependencies and contextual nuances in customer reviews, making it a powerful model for sentiment classification tasks. By comparing the performance of these models, this study aims to identify the most effective approach for predicting review sentiment polarity and to provide actionable insights for businesses seeking to improve their understanding of customer opinions.

In this paper, we utilize both traditional Supervised Machine Learning algorithms and advanced Recurrent Neural Network (RNN) models to predict the sentiment polarity of Amazon product reviews. Supervised learning techniques allow us to train models on label datasets to accurately classify sentiments based on extracted features. RNN models, particularly LSTM, enable us to capture the sequential dependencies and contextual meaning within the review text, providing deeper insight into customer opinions.

II. LITERATURE REVIEW

Previous research on review helpfulness has primarily concentrated on automating the prediction of helpfulness to manage the vast volume and inconsistent quality of online consumer reviews. Much of the work has explored the relationship between text-based features and helpfulness using text mining and statistical models.

Salehan et al. (2016) examined the influence of sentiment polarity on review helpfulness, considering elements such as review length, title, and longevity. Their findings indicated that neutral sentiment reviews significantly impacted helpfulness, and longer reviews tended to be more informative and thus more helpful.

Cao et al. (2011) explored why certain reviews receive more helpfulness votes than others, focusing on basic, stylistic, and semantic characteristics using data from CNET Download.com. They employed an Ordinal Logistic Regression model and evaluated performance using metrics like misclassification rate and AIC. Their findings highlighted that semantic features were the most influential, and reviews expressing extreme opinions tended to receive more helpfulness votes. Similarly, Liu et al. (2008) used a non-linear regression model on IMDB reviews, emphasizing factors like reviewer expertise, writing style, and timeliness. These features proved most significant when combined, as indicated by lower MSE scores.

Other notable studies include Kim et al. (2006), who focused on providing real-time feedback on review helpfulness using SVM regression, identifying review length, unigrams, and product ratings as key factors. Ghose et al. (2007) addressed the challenge of handling massive volumes of online reviews by building ranking mechanisms that incorporated both user- and manufacturer-oriented perspectives. Their findings showed that reviews combining subjective and objective elements were considered more helpful, demonstrating the effectiveness of hybrid analysis approaches for evaluating online reviews.

III. PROBLEM STATEMENT

Manually interpreting vast numbers of product reviews is time-consuming and inconsistent. This necessitates the development of an intelligent system that can predict the sentiment polarity of reviews (positive or negative) with high accuracy using machine learning and deep learning methods.

IV. DATA SET DESCRIPTION

The dataset used for this project was obtained from Kaggle. It consists of product reviews collected over a span of 13 years, from October 1999 to October 2012. The goal is to predict the sentiment polarity of a given review—classifying it as positive if the rating is 4 or 5, and negative if the rating is 1 or 2.

A. Data Includes

1) Number of reviews: 568,454
2) Number of users: 256,059
3) Number of products: 74,258
4) Timespan: Oct 1999 - Oct 2012
5) Number of Attributes/Columns in data: 10

B. Attribute Information

1) Id: unique Numeric data for every data
2) Product Id: unique Alphanumeric data for every product
3) User Id: unique Alphanumeric data for every user
4) Profile Name: Name of the user
5) Helpfulness Numerator: Number of people who found the review helpful
6) Helpfulness Denominator: Number of people who indicated whether or not review is helpful
7) Score: total number of ratings from 1 to 5.
8) Time: Timespan of reviews
9) Summary: textual data about the product (Briefly)
10) Text: textual data about the product (Review)

V. METHODOLOGIES

This research follows a structured approach including data acquisition, preprocessing, feature engineering, model training, and evaluation. Both traditional ML classifiers and an LSTM neural network are trained using different vector representations like Bow, TF-IDF, and Word2Vec (CBOW, Skip Gram). The methodologies followed in this study are outlined below.

1) Response Variable Generation
2) Text Preprocessing
3) Training Models for Classification
4) Model Validation

VI. RESPONSE VARIABLE GENERATION

The dataset includes Amazon product reviews with attributes such as Review Text, Score, Helpfulness metrics, and Summary. Reviews with score > 3 as positive, and < 3 as negative. Neutral reviews are excluded for binary classification, because this study is focusing only on Binary Classification.

VII. PREPROCESSING TECHNIQUES

The Data pre-processing is an important step in our analysis because it has a significant impact on the results. An unprocessed dataset can produce incorrect results and affect the analysis; therefore, pre-processing the data is required before performing any data mining operation. Before data preprocessing, we must deduplicate our data. Now, To Preprocess the data, the following steps were taken:

1) Begin by removing the html tags
2) Remove any punctuations or limited set of special characters like , or . or # etc.
3) Check if the word is made up of English letters and is not alpha-numeric
4) Check to see if the length of the word is greater than 2 (as it was researched that there is no adjective in 2-letters)
5) Convert the word to lowercase
6) Remove Stop words
7) Finally Snowball Stemming the word (it was observed to be better than Porter Stemming)

VIII. FEATURE EXTRACTION METHODS

Multiple vectorization techniques were explored to represent text data numerically for all Machine Leaning Algorithms mentioned above. The methods include:

1) Bag of Words (Bow - Unigram and Bigram)
2) TF-IDF (Unigram and Bigram)
3) Word2Vec (Average and TF-IDF weighted)

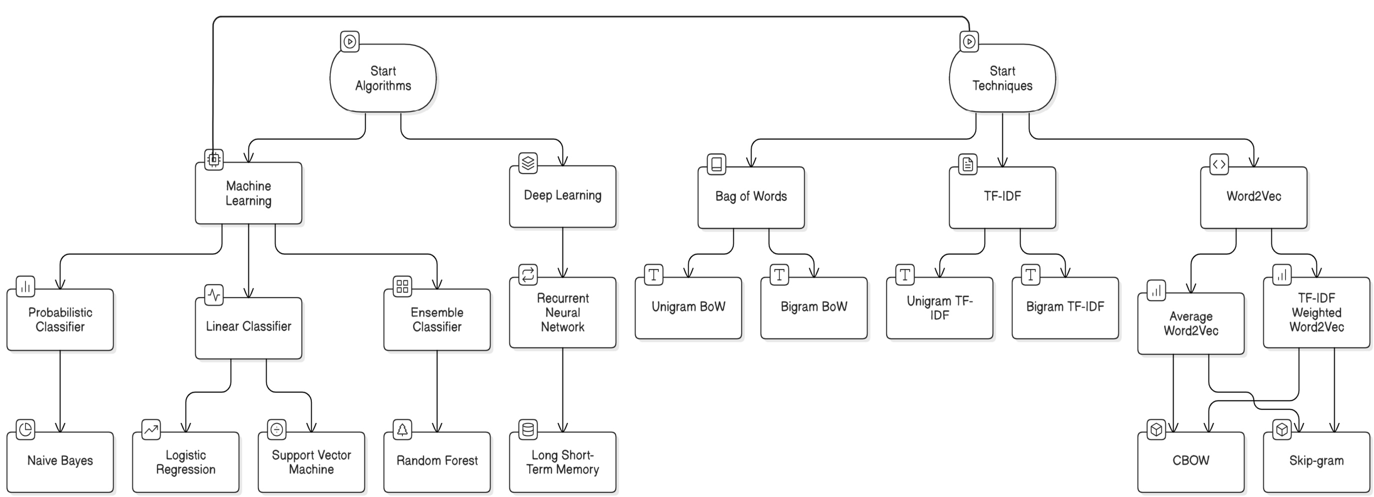


Fig 8.1: Text Classification Pipeline: Algorithms and Vectorization Technique

IX. MACHINE LEARNING MODELS

A range of machine learning models were implemented using scikit-learn and Keras. The classifiers used include:

1) Naive Bayes: Fast and simple, but limited in performance.
2) Logistic Regression: Balanced and widely used for linear classification.
3) Support Vector Machines: Effective in high-dimensional space.
4) Random Forest: Ensemble learning method that handles feature interactions well.

Model performance was primarily evaluated using the Area Under the ROC Curve (AUC) through 10-fold cross-validation on an 80/20 train-test split. Grid search was employed to optimize hyperparameters for SVM and Random Forest models.

X. DEEP LEARNING MODELS

Long Short-Term Memory (LSTM) is a type of Recurrent Neural Network capable of learning long-term dependencies. It processes sequences and retains contextual meaning, making it ideal for sentiment analysis tasks. The LSTM model was trained using pre-processed text sequences and word embeddings. It outperformed other models in terms of capturing nuanced sentiments and maintaining context across longer reviews.

XI. RESULTS AND EVALUATION

The models were evaluated using metrics such as Accuracy, Precision, Recall, and AUC-ROC.

1) Naive Bayes

The Naive Bayes algorithm is concerned with the probability of events. It is a classifier based on probability. This classifier is based on the Bayes Theorem

P(class | data) = (P(data | class) * P(class)) / P(data)

Naive Bayes gives a baseline model. In this algorithm, the best estimate hyperparameter for alpha is 0.1. As a result, the Train AUC and Test AUC are below across different vector representations.

Vector	Algorithm	Hyperparameter-alpha	Train AUC	Test AUC
bow	naive-bayes	0.1	0.9452974063624253	0.903594654994184
bigram_bow	naive-bayes	0.1	0.9997501437485466	0.9081245741461433
tfidf	naive-bayes	0.1	0.9547456479504195	0.916519427182046
tfidf_bigram	naive-bayes	0.1	0.9995902613135251	0.9248475892556587

Figure 11.1: AUC Scores of Naïve Bayes Algorithm with Different Text Vectorization Techniques

Key Findings – Naïve Bayes Models

- Bow + Naïve Bayes gave good performance with Test AUC of 0.90.
- Bigram Bow + Naïve Bayes slightly improved the Test AUC to 0.91.
- TF-IDF + Naïve Bayes showed better results with Test AUC of 0.91.
- Bigram TF-IDF + Naïve Bayes gave the best Test AUC of 0.92, showing strong generalization.

2) Logistic Regression

Logistic regression is a method for predicting a binary outcome: either something happens or nothing happens. The dependent variable must be categorical, but the independent variable might be categorical or numeric. As it is written:

- P(Y=1|X) or P(Y=0|X)
- In this algorithm, the best estimate hyperparameter for C is 1000 and the penalty is l2 regularization. As a result, the
- As a result, the Train AUC and Test AUC are below across different vector representations.

Vector	Algorithm	regulariation	Hyperparameter-C	Train AUC	Test AUC
bow	logistic regression	l2	0.1	0.9620248079612217	0.9388352606745828
bigram_bow	logistic regression	l2	1	0.9999882556538641	0.9482435529787344
tfidf	logistic regression	l2	10	0.976282703805722	0.9545681039988041
bigram_tfidf	logistic regression	l2	1000	0.9999999982001002	0.9615304946357388
avg-word2vec	logistic regression	l1	1000	0.9124118837008574	0.9107224838650744
skpgm_avg-word2vec	logistic regression	l1	1000	0.9215686520496108	0.9202296960948522
tfidf-word2vec	logistic regression	l2	1	0.8849181988562624	0.8834473563212578
skpgm_tfidf-word2vec	logistic regression	l2	1000	0.8904718648070211	0.8883955599147808

Figure 11.2: AUC Scores of Logistic Regression Algorithm with Different Text Vectorization Techniques

Key Findings – Logistic Regression Models
- Bow + Logistic Regression gave a solid Test AUC of 0.93.
- Bigram Bow + Logistic Regression improved further to 0.94 Test AUC.
- TF-IDF + Logistic Regression performed well with 0.95 Test AUC.
- Bigram TF-IDF + Logistic Regression gave the highest Test AUC of 0.96.
- Average Word2Vec + Logistic Regression gave stable results with 0.91 Test AUC.
- Skip-gram Word2Vec + Logistic Regression slightly improved to 0.92 Test AUC.
- TF-IDF Weighted Word2Vec models gave lower AUCs around 0.88–0.89.

3) Support Vector Machine

Stochastic Gradient Descent (SGD) is a simple yet very efficient approach to fitting linear classifiers and regressors under convex loss functions like (linear) Support Vector Machines.

- Here We are using SGD Classifier. In this algorithm, the best estimate hyperparameter for alpha is 0.0001 and penalty is l2. As a result, the Train AUC and Test AUC are below across different vector representations.

Vector	Algorithm	kernel	penalty	Hyperparam-alpha	Hyperparam-C	gamma	Train AUC	Test AUC
bow	SVM	linear	l2	0.001	-	-	0.9502190064968571	0.9379670838643561
bigram_bow	SVM	linear	l2	0.001	-	-	0.9836558092595505	0.9508929156110826
tfidf	SVM	linear	l2	0.0001	-	-	0.9520451785250947	0.9449439641821381
bigram_tfidf	SVM	linear	l2	0.0001	-	-	0.9672658970146368	0.9475885251141982
avg-w2v	SVM	linear	l2	0.001	-	-	0.9115496939031019	0.9099053863665842
skip_gram_avg-w2v	SVM	linear	l2	0.001	-	-	0.9204305299643304	0.9193060645459938
tfidf-w2v	SVM	linear	l2	0.001	-	-	0.8835289898501959	0.8819927326658117
skipgram_tfidf-w2v	SVM	linear	l2	0.0001	-	-	0.8884426721819147	0.8865317750483183

Figure 11.3: AUC Scores of SVM Classifier (SGD-based) with Different Text Vectorization Techniques

Key Findings – SVM Models
- Bow + SVM gave a solid performance with Test AUC of 0.93.
- Bigram Bow + SVM showed improved performance with Test AUC of 0.95.
- TF-IDF + SVM also performed well with Test AUC of 0.94.
- Bigram TF-IDF + SVM performed best among SVMs with Test AUC of 0.94+.
- Average Word2Vec + SVM gave Test AUC of 0.90, showing decent performance.
- Skip-gram Word2Vec + SVM slightly better with Test AUC of 0.91.
- TF-IDF Weighted Word2Vec + SVM models gave lower AUCs around 0.88.

4) Random Forest

Random Forest classifier is a meta-estimator that fits a number of decision trees on various subsamples of datasets and uses average to improve the predictive accuracy of the model and controls over-fitting. In this algorithm, the best estimate hyperparameter for max-depth is 25 and the n-estimator is 120. As a result, the Train AUC and Test AUC are below across different vector representations.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue V May 2025- Available at www.ijraset.com

Vector	Algorithm	Hyperparam-n_estimator	Hyperparam-max_Depth	Train AUC	Test AUC
bow	random forest	120	25	0.9658901787074783	0.9167058103234019
bigram_bow	random forest	120	30	0.9620760164601987	0.919630563941423
tfidf	random forest	120	30	0.98123587008918	0.9215174128932738
bigram_tfidf	random forest	120	25	0.958668905375427	0.9180772567705946
avgw2v	random forest	120	25	0.9999999275540334	0.9046545788859606
skipgram_avgw2v	random forest	120	25	0.9999998425087684	0.9130353808239032
tfidfw2v	random forest	120	30	0.9999991616966725	0.8803284743703197
skipgram_tfidfw2v	random forest	120	30	0.9999989484085472	0.882164535542237

Figure 11.4: AUC Scores of Random Forest Classifier with Different Text Vectorization Techniques

Key Findings – Random Forest Models
• Bow + Random Forest gave good results with Test AUC of 0.91.
• Bigram Bow slightly better with Test AUC of 0.92.
• TF-IDF performed best with Test AUC of 0.92+.
• Bigram TF-IDF showed similar performance at 0.91+ AUC.
• Average Word2Vec gave high train AUC but Test AUC dropped to 0.90, suggesting overfitting.
• Skip-gram Word2Vec improved a bit with Test AUC of 0.91.
• TF-IDF Weighted Word2Vec models had lower Test AUC around 0.88.

5) *Long Short-Term Memory (LSTM)*

LSTM is a type of recurrent neural network (RNN) capable of learning long-term dependencies. It is particularly effective for sequence-based tasks like sentiment analysis, as it retains context over input sequences. In this study, LSTM models were trained using word embeddings from the Keras Embedding layer to learn distributed representations of words directly from the data. Two models were tested with varying LSTM units and dropout values.

The models were trained using the Adam optimizer and binary cross-entropy loss, with dropout applied after the embedding layer and the LSTM layer to reduce overfitting. Below are the configurations and results of the LSTM models across embedded vector representations:

Vector	Algorithm	Hyperparameters	Train AUC	Test AUC
sequence (embedded)	LSTM	LSTM=100, Dropout=0.6/0.4, Epochs=10, Batch=128	0.9889	0.9312
sequence (embedded)	LSTM	LSTM=70, Dropout=0.6/0.5, Epochs=12, Batch=128	0.9907	0.9296

Figure 11.5: AUC Scores of LSTM Classifier with Embedded Word Vectors

Key Findings – LSTM Models

- Sequence Embedding + LSTM (100 units) delivered the best performance with a Test AUC of 0.9312, outperforming traditional vectorization-based models, indicating strong generalization capabilities.
- Both LSTM models achieved high Train AUC values (~0.99), demonstrating the models' ability to learn complex patterns in the training data.
- Overfitting was well-controlled using appropriate dropout settings (0.6 at embedding and 0.4/0.5 after LSTM), contributing to robust test performance.
- Model with fewer LSTM units (70) showed slightly lower Test AUC (0.9296) but comparable generalization, indicating that even shallower LSTM networks can perform well with regularization.
- LSTM models surpassed all Random Forest models, including those based on TF-IDF and Word2Vec, in terms of Test AUC, proving the strength of deep sequence modelling.
- The use of Keras Embedding layer allowed the model to learn semantic representations specific to the sentiment classification task, enhancing performance over static embeddings like Word2Vec.

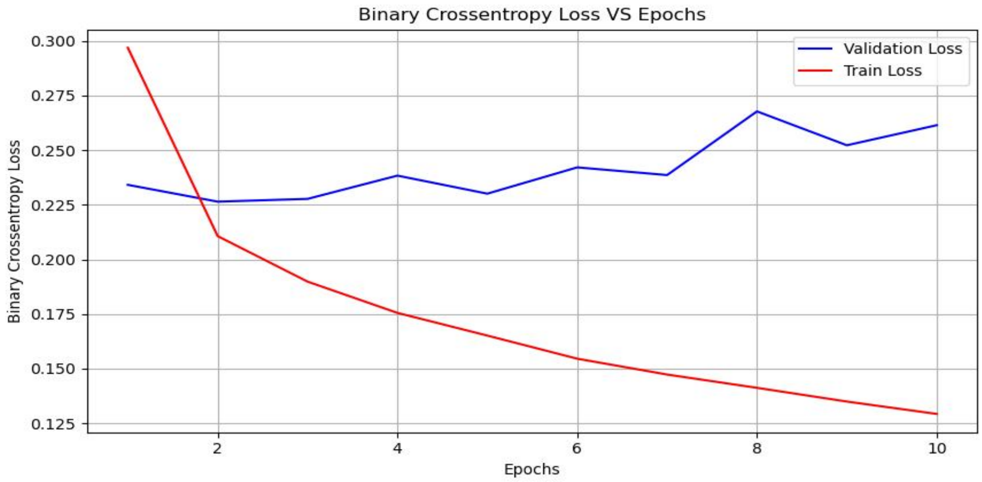


Figure 11.6: Binary Crossentropy Loss of LSTM Model Across Epochs on Training and Validation Data for LSTM (100) units

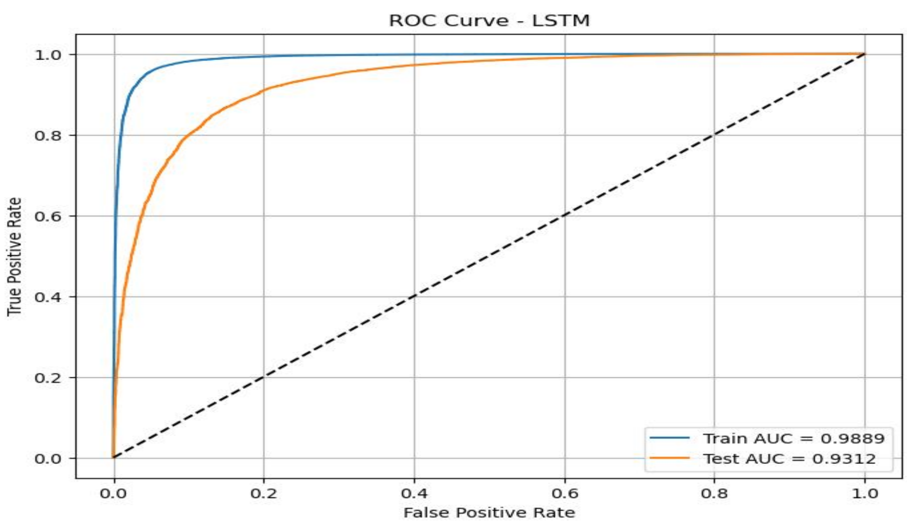


Figure 11.7: ROC Curve and AUC Scores of LSTM Model on Training and Test Data for LSTM (100) units

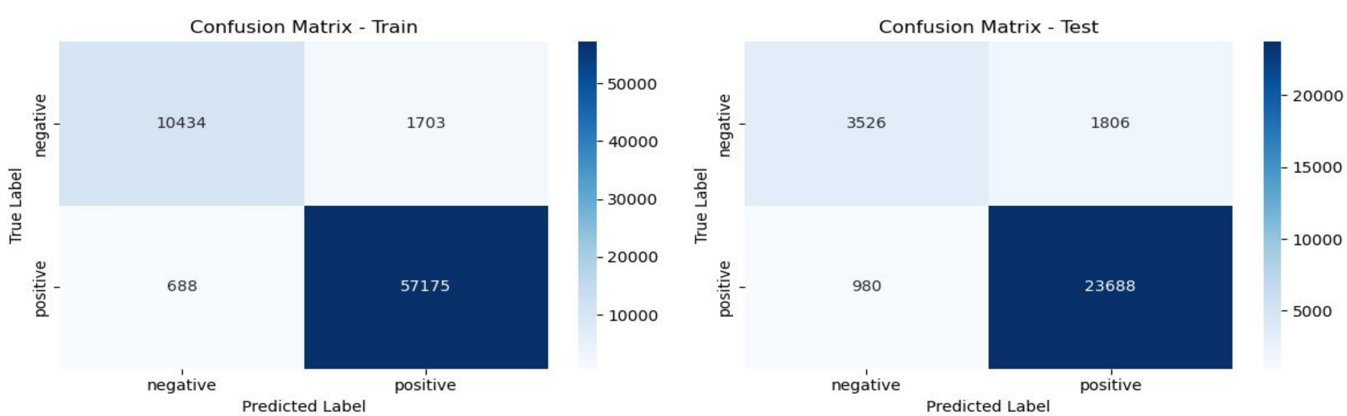
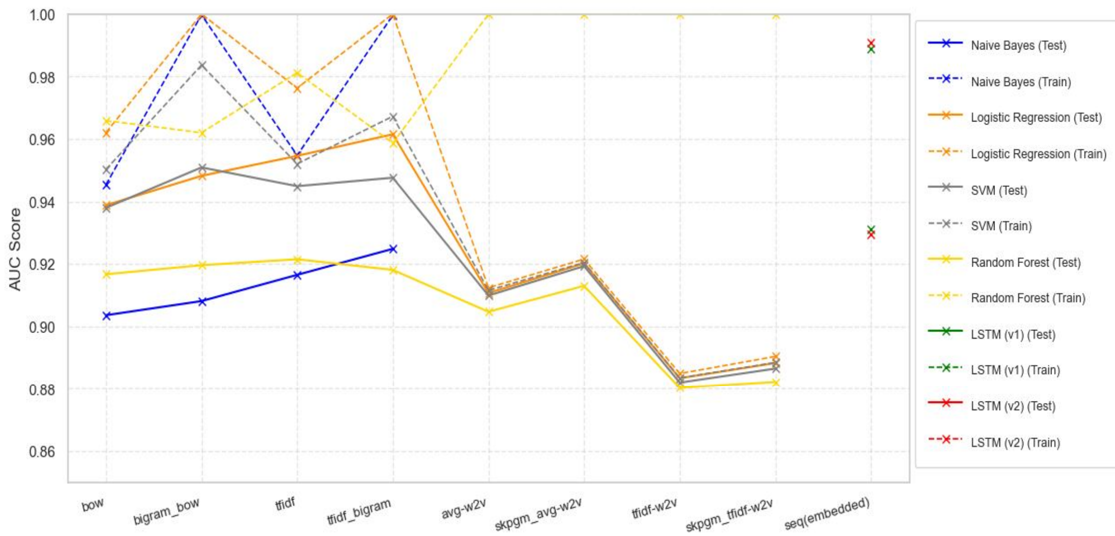


Figure 11.8: Confusion Matrices of LSTM Model on Training and Test Sets for LSTM (100) units

6) Comparison of Features and Classifiers used in the Study

The figure presents a detailed comparison of AUC scores for several machine learning and deep learning models applied to different text feature extraction techniques. Models such as Naive Bayes, Logistic Regression, SVM, and Random Forest were evaluated using features like Bag-of-Words, TF-IDF, and Word2Vec variants. Among the traditional models, Logistic Regression and SVM performed consistently well, particularly with bigram and TF-IDF-based features, achieving high AUC scores.

Deep learning models, specifically two versions of LSTM, were evaluated using embedded sequential inputs. LSTM(100) units demonstrated the best test AUC score of 0.9312, showing its strong capability to capture sequential dependencies in the text.



	bow(Tr, Te)	bigram_bow(Tr, Te)	tfidf(Tr, Te)	tfidf_bigram(Tr, Te)	avg-w2v(Tr, Te)	skpgm_avg-w2v(Tr, Te)	tfidf-w2v(Tr, Te)	skpgm_tfidf-w2v(Tr, Te)	seq(embedded)(Tr, Te)
Naive Bayes	0.9453, 0.9036	0.9998, 0.9081	0.9547, 0.9165	0.9996, 0.9248					
Logistic Regression	0.9620, 0.9388	1.0000, 0.9482	0.9763, 0.9546	1.0000, 0.9615	0.9124, 0.9107	0.9216, 0.9202	0.8849, 0.8834	0.8904, 0.8883	
SVM	0.9502, 0.9380	0.9837, 0.9509	0.9520, 0.9449	0.9673, 0.9476	0.9115, 0.9099	0.9204, 0.9193	0.8835, 0.8820	0.8884, 0.8865	
Random Forest	0.9659, 0.9167	0.9620, 0.9196	0.9812, 0.9215	0.9587, 0.9181	0.9999, 0.9047	0.9999, 0.9130	0.9999, 0.8803	0.9999, 0.8822	
LSTM (v1)									0.9889, 0.9312
LSTM (v2)									0.9907, 0.9296

Figure 11.9: Performance comparison of classifiers used in the study

Overall, the results suggest that while traditional models perform well with hand-crafted features, LSTMs with embedded inputs offer superior generalization for sentiment analysis tasks.

XII. CONCLUSION AND FUTURE WORK

The comparative evaluation of Naive Bayes, Logistic Regression, SVM, Random Forest, and LSTM models reveals diverse patterns of performance depending on the vectorization strategy used. Naive Bayes, when used with Bag of Words, showed a reliable Test AUC of 0.9036 without overfitting, indicating solid performance in text-based tasks. Logistic Regression excelled when paired with bigram TF-IDF vectors, achieving a Test AUC of 0.9615, and also demonstrated high robustness with TF-IDF (0.9546). Although Average Word2Vec-based embeddings provided decent results (~0.91, 0.92 AUC), their performance slightly lagged in count-based and skip gram based Word2Vec TF-IDF models in this setup.

SVM models, known for their margin-based classification, delivered strong generalization, especially with bigram TF-IDF (Test AUC of 0.9476) and unigram TF-IDF (0.9449), confirming their effectiveness in high-dimensional sparse spaces. Word2Vec-based features with SVM also performed well but were slightly behind in Test AUC compared to TF-IDF approaches.

Random Forest classifiers achieved excellent Train AUCs across the board, indicating strong learning capability, but some configurations showed signs of overfitting, particularly with Word2Vec embeddings (Train AUC ≈ 1.0 vs Test AUC < 0.92). The best performance from Random Forest was with TF-IDF vectors (Test AUC of 0.9215) and bigram Bow (0.9196), confirming the strength of count-based representations in tree-based models.

Introducing LSTM, a deep learning model designed for sequence learning, significantly enhanced performance, particularly in capturing contextual dependencies and nuanced sentiment expressions. With learned embeddings and dropout regularization, LSTM achieved Test AUCs of 0.9312 and 0.9296 for 100 and 70 LSTM units respectively, outperforming all classical models. This demonstrates the advantage of deep learning in handling complex sentence structures and long-term dependencies in textual data.

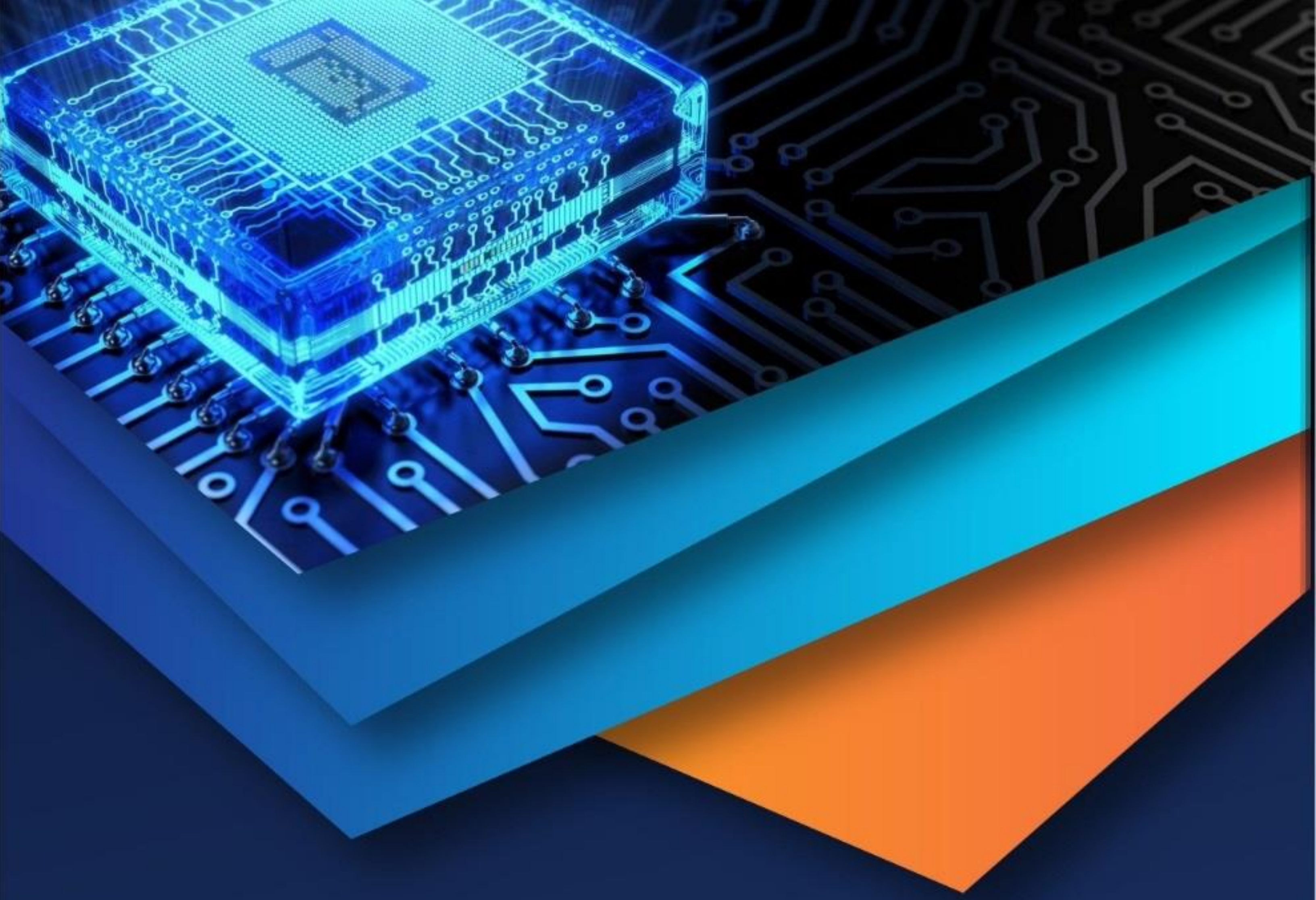
To push the boundaries further, integrating transformer-based models like BERT could yield even more contextually aware predictions due to its bidirectional attention mechanism, which has shown state-of-the-art performance in numerous NLP tasks. Expanding the current system to handle multilingual and code-mixed reviews would improve adaptability in global markets. Additionally, deploying these models into real-time sentiment analysis dashboards can provide actionable insights to e-commerce businesses, enabling dynamic customer feedback analysis, product improvement, and enhanced decision-making.

XIII. ACKNOWLEDGMENT

With deep appreciation, we would like to thank everyone who supported and contributed to the successful completion of this study. We are especially grateful to Sri Venkateswara College of Engineering & Technology (SVCET), Chittoor, Andhra Pradesh, India for providing the academic environment and resources essential for carrying out this research. We express our heartfelt thanks to Mr. M. Navalan, and all faculty members for their continuous guidance, encouragement, and support throughout the project. Their contributions and constructive feedback played a vital role in improving the depth and quality of our research work.

REFERENCES

[1] Pang, B., & Lee, L. (2008). Opinion mining and sentiment analysis.
[2] Mikolov, T., et al. (2013). Distributed Representations of Words and Phrases and their Compositionality.
[3] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory.
[4] Akanksha Halde, Aditi Uttekar & Amit Vishwakarma, “sentiment analysis on amazon product reviews” , Volume:04/Issue:04/April-2022, e-ISSN: 2582-5208
[5] Kartikay Thakkar, Sidharth Sharma, Ujjwal Chhabra & Asst. Prof. Ms. Charu Gupta, “sentimental analysis on amazon fine food reviews”, Journal of International Journal of Scientific Research & Engineering
[6] Lilleberg, J., Zhu, Y. and Zhang, Y., 2015, July. Support vector machines and word2vec for text classification with semantic features. In 2015 IEEE 14th International Conference on Cognitive Informatics & Cognitive Computing (ICCI* CC) (pp. 136-140). IEEE.
[7] Sokolova, M. and Lapalme, G., 2009. A systematic analysis of performance measures for classification tasks. Information Processing & Management, 45(4), pp.427-437
[8] Hsu, C.W., Chang, C.C. and Lin, C.J., 2003. A practical guide to support vector classification.
[9] Salehan, M. and Kim, D.J., 2016. Predicting the performance of online consumer reviews: A sentiment mining approach to big data analytics. Decision Support Systems, 81, pp.30-40.
[10] Challenges for NLP Frameworks. pp. 45-50. Valletta, Malta, May 2010. ELRA. http://is.muni.cz/publication/884893/en.
[11] Rong, X., 2014. word2vec parameter learning explained. arXiv preprint arXiv:1411.2738.



10.22214/IJRASET

INDEX COPERNICUS

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY