International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

Classification of Amazon Product Reviews Using Hybrid Embedding Techniques for Sentiment Analysis

G Dinesh[1], D Nagaraj[2]

[1]P.G Scholar, Department of MCA, Sri Venkatesa Perumal College of Engineering & Technology, Puttur.
E-mail: gettinidinesh@gmail.com, ORC-ID: https://orcid.org/0009-0008-6107-2725

[2]Professor, Department of CSE, Sri Venkatesa Perumal College of Engineering & Technology, Puttur.
E-mail: raj2dasari@gmail.com, ORC-ID: https://orcid.org/0000-0002-8511-1863

Abstract: Sentiment Analysis (SA) is an important part of Natural Language Processing (NLP) for getting views and feelings out of text data so that it can be correctly categorized and decisions can be made based on that information. A key part of improving prediction performance is making sure that features are represented correctly. The study uses Kaggle's Amazon review dataset, which has 4,000,000 records but was reduced to just 50,000 reviews for speed reasons. The dataset has a variety of product reviews that have been labeled with different types of emotion. Normalization, HTML/URL removal, tokenization, label encoding, and n-gram creation are all parts of preprocessing. Sentiment distribution and word clouds help you understand features. Several embedding methods are used to make feature vectors, such as Bag-of-Words (BoW), TF-IDF, Word2Vec, BERT, and a model that combines BERT and TF-IDF. Accuracy, precision, recall, F1-score, and confusion matrices are used to train and test machine learning models like Naive Bayes, KNN, Decision Tree, Random Forest, and Linear SVC. To make things even easier to understand and use, advanced models like XGBoost and ElectraModel are used with mixed embeddings that combine BERT, TF-IDF, and BoW. Explainable AI methods, like LIME and SHAP, help us figure out how important a feature is. Real-time sentiment prediction and topic models are possible with a Flask-based interface. When used with ElectraModel, Electra embeddings perform better than baseline and conventional embeddings, hitting 91.1% accuracy.

"Index Terms: *BERT, classification, feature extraction, machine learning, sentiment analysis, Word embeddings, Word2Vec".*

1. INTRODUCTION

People all over the world can share their thoughts on a wide range of topics and events thanks to the widespread use of mobile apps and social networking sites [1]. People can write reviews and give comments about goods and services on sites like Facebook, Twitter, and e-commerce sites [2]. Online shopping is becoming more and more popular, which has made the internet an important place for people to voice their views [3]. A lot of retail websites actively ask customers to write reviews. This creates a large database of user-generated content that is useful for both buyers and sellers. People who are thinking about buying often use these reviews to help them make smart choices. Companies that make products also use them to make their marketing strategies and new products better.

Opinion Mining (OM), which is sometimes called Sentiment Analysis (SA), looks at random text and sorts subjective information into groups instead of finding topics [4]. Machine learning (ML) methods are very important in OM because they let systems learn from past data to make opinion classification more accurate. One of the biggest problems with ML-based OM, though, is getting raw data and turning it into features that can be used for models [5]. The process of feature engineering turns unstructured data into representations that make sense. This makes models more accurate and improves their total performance. Feature extraction turns unstructured text into numerical vectors in natural language processing (NLP), which lets machine learning algorithms handle the data well [6].

Bag-of-Words (BoW), Term Frequency-Inverse Document Frequency (TF-IDF), and N-grams are some of the most common ways to identify features [7]. The BoW model encodes text by the number of times a word appears, without taking into account the order or meaning of the words [8]. TF-IDF is better than BoW because it ranks words by how important they are in a text compared to the corpus. This makes unique words stand out more while making common words less important. However, both BoW and TF-IDF are based on numbers that are based on frequency and don't take into account how words relate to each other in a semantic way.

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

When you look at a list of n-grams, which are groups of n words, they show how words come together in context, which helps you understand what they mean [9].

New ways of representing features, like word embeddings, get around these problems by turning words into dense vector spaces that show how similar they are in meaning [10]. Using designs like Continuous Bag-of-Words (CBoW) and Skip-Gram, models like Word2Vec create embeddings that let related words get similar vector representations. More advanced NLP apps, like sentiment analysis, text classification, and machine translation, use these embeddings as a base. They give richer and more informative representations than older methods.

2. LITERATURE REVIEW

Recent progress in natural language processing (NLP) and machine learning (ML) has made it possible for mood analysis and opinion mining from text data to get a lot better. In their study [11], Kobra et al. looked into multihead text mining methods for studying COVID-19 feedback using machine learning, deep learning, and hybrid deep learning models. Their research focused on how traditional machine learning algorithms can be combined with deep learning architectures to effectively get useful information from a lot of user feedback. It showed that hybrid approaches are better at accuracy and feature representation than either ML or deep learning techniques used alone. This study showed that combining different types of analysis can help with sentiment classification, especially in areas where text data is complicated and nuanced.

Yang and Ding [12] worked on improving word embedding learning by adding better distance weighting and changing the size of the schedule window. Their method got around problems with regular Word2Vec models, like how static window sizes and equal distance factors don't always pick up on subtle differences in meaning between words. The suggested way improved the semantic quality of embeddings and made it easier to do NLP tasks like text classification and sentiment analysis by changing the window size and distance weights on the fly. This study showed how important it is to fine-tune embedding generation processes so that they better show how words relate to each other in context.

Rezende et al. [13] looked into how NLP methods could be used with Latent Semantic Analysis (LSA), Word2Vec, and Word Mover's Distance (WMD) to predict new technologies and find similarities in patent documents. Their research showed that using both statistical and neural embedding models together can better capture meaning similarity than using either one alone. The study focused on how using more than one feature extraction technique together improves performance in tasks that need accurate similarity calculations. This gives us ideas for how to make embedding methods work better for specific uses.

Mir et al. [14] suggested a way to find fake online reviews using supervised machine learning methods along with the BERT model. While the study found that standard supervised classifiers could somewhat spot fake reviews, adding BERT embeddings made classification much better. The authors showed that context-aware embeddings are important for finding minor linguistic cues that point to fake content by fine-tuning BERT on domain-specific review datasets. This shows how important transformer-based models are becoming in opinion mining.

González-Carvajal and Garrido-Merchán [15] looked at how BERT stacked up against more standard machine learning text classification methods. Their results showed that BERT constantly did better than common models like SVM, Random Forest, and Naïve Bayes in a number of text classification tests. This was because it understood context in both directions. The study also pointed out that transformer models that have already been trained can be quickly tweaked to work better in certain areas, allowing for better syntactic understanding without a lot of feature engineering.

Devlin et al. [16] created the first BERT design, which is a deep bidirectional transformer model that has already been trained to understand language. It was BERT who changed NLP by adding attention-based systems that could learn context from both the left and right parts of a token. This design made it possible to do better at many NLP tasks, such as answering questions, recognizing named entities, and analyzing sentiment. Zou and Wang [17] suggested a semi-supervised short text mood classification method that added to BERT's abilities. Because there weren't many labeled samples available, their model used unlabeled data to improve learning. The study showed that BERT-

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

based designs can work well in semi-supervised settings, keeping their high level of classification accuracy even when there aren't many annotations.

In his paper [18], Horev gave a thorough introduction to BERT, describing how it works and how it can be used in NLP. The piece explained the purpose of attention mechanisms, tokenization, and contextual embeddings. This made BERT easier for more people to understand and reinforced its use in sentiment analysis. Nozza et al. [19] looked into language-specific BERT models and compared how well general and language-adapted versions worked. Their research showed that domain- or language-specific pre-training makes models much better at knowing localized semantics. This shows how important it is to use customized pre-training strategies for applications that use more than one language or domain.

The study by Belguith et al. [20] was mostly about aspect-level sentiment analysis. They used deep learning techniques along with ontological knowledge to get sentiments related to certain product characteristics. Their method showed that combining structured subject knowledge with neural models can make sentiment classification easier to understand and more accurate, especially when dealing with reviews that aren't straightforward. By using ontologies, the model could clear up any confusion in textual expressions of sentiment, giving customers and producers more useful information.

3. MATERIALS AND METHODS

Using advanced natural language processing and machine learning methods, the suggested system figures out how people feel about Amazon product reviews. A Kaggle dataset with 4,000,000 records is selected to show 50,000 reviews from a wide range of product categories that have been labeled with mood classes. Text normalization, HTML and URL removal, tokenization, label encoding, and n-gram creation are all parts of preprocessing. We get feature vectors from Bag-of-Words (BoW), Term Frequency-Inverse Document Frequency (TF-IDF), Word2Vec, BERT embeddings, and a model that combines BERT and TF-IDF. A number of classifiers are trained with these embeddings, such as Naive Bayes, K-Nearest Neighbors (KNN), Decision Tree, Random Forest, Linear Support Vector Classifier (SVC), XGBoost, and ElectraModel. AI techniques that can be explained, like LIME and SHAP, give us information about how important different features are. A Flask-based interface lets us predict sentiment in real time and work with topic modeling, combining many embeddings and classifiers for strong, understandable analysis.

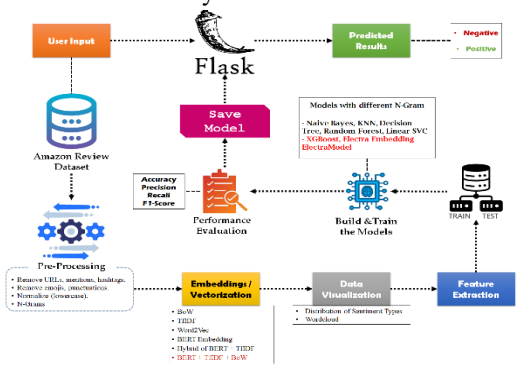

Fig.1 Proposed Architecture

For sentiment analysis, the system design includes modules for preprocessing data, extracting features, training models, and making predictions in real time. Normalizing, removing HTML and URLs, tokenizing, and making n-grams are all things that are done to raw Amazon reviews. BoW, TF-IDF, Word2Vec, BERT, and mixed embeddings are all used to make feature representations. These features are used to teach machine learning classifiers like Naive Bayes, KNN, Decision Tree, Random Forest, SVC, XGBoost, and ElectraModel. Explainable AI tools like LIME and SHAP make data easy to understand, and a Flask-based interface lets you do things like live visualization, sentiment prediction, and topic modeling, which makes sure that the analysis is strong and clear.

a) Dataset Collection:

The dataset is made up of 50,000 randomly chosen Amazon product reviews from a Kaggle dataset with 4,000,000 records. There are three columns in it: text, name, and title. There is one missing title entry in the dataset, but it has 25,007 good reviews (label 1) and 24,993 negative reviews (label 2). Each record is a user review with a mood label that goes with it. This creates a balanced set of textual data that can be used to train and test sentiment analysis models.

	label	title	text
0	2	This is a great book	I must preface this by saying that I am not re...
1	1	Huge Disappointment.	As a big time, long term Trevanian fan, I was ...
2	2	Wayne is tight but cant hang with Turk.	This album is hot as it wants to be. However C...
3	2	Excellent	I read this book when I was in elementary scho...
4	1	Not about Anusara	Although this book is touted on several Anusar...

Fig.2 Amazon product reviews

b) Pre-Processing:

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

Text material is cleaned in a lot of different ways to make sure it is consistent and of good quality. This includes normalization, which gets rid of HTML tags, URLs, punctuation, and other characters that aren't needed. This makes the data less noisy. Tokenization separates the text into single words so that they can be analyzed, and label encoding turns emotion labels into numbers. N-gram generation finds patterns of words that appear in a certain order. This gets the dataset ready for embedding methods and makes sure it works with machine learning models.

c) Embeddings / Vectorization:

Feature extraction turns written data into numbers that can be used by machines to learn. Some of the techniques used are Bag-of-Words (BoW), TF-IDF, Word2Vec, BERT embeddings, and mixed BERT + TF-IDF mixes. These methods store both statistical and semantic data. BoW and TF-IDF measure how often and how important words are, while Word2Vec and BERT measure how words fit into their surroundings. Hybrid embeddings combine strengths that work well together to make rich feature vectors that help models understand the meaning and structure of text better.

d) Data Visualization:

Visual analysis helps us understand the text information and the patterns of emotion. Sentiment distribution plots show the balance of the information by showing the percentage of positive and negative reviews. Word clouds show the most popular words, which can help you figure out what people are talking about and how they talk. These pictures help find patterns, spot biases, and choose which features to use. They give you a better idea of what the dataset is about, which helps you make better choices during feature engineering and model development.

e) Feature Extraction:

Text that has been processed is turned into organized feature sets that store both information based on frequency and information based on context. Statistical features like BoW and TF-IDF show how often and how important a word is, while embeddings like Word2Vec and BERT show how words relate to each other in a logical way. This makes sure that models get a lot of useful data, which speeds up learning and makes predictions more accurate. Feature extraction is the first step in training, which makes it possible for models to tell the difference between different emotions.

f) Training and Testing:

The cleaned and vectorized dataset is split into training and testing groups to make evaluating the model easier. Training data is used to shape machine learning models, which helps them find patterns in how people share their feelings. Testing data checks how well models work on reviews they haven't seen before, making sure that the evaluation is fair. This separation helps avoid overfitting, checks for robustness, and makes sure that models for predicting mood stay accurate across a wide range of Amazon review data.

g) Algorithms:

Naive Bayes: Uses probabilistic reasoning and the idea that features are independent to handle big text datasets well. Used BoW, TF-IDF, Word2Vec, BERT, and hybrid embeddings along with unigram to trigram n-grams to find phrase-level dependencies and contextual semantics for classifying mood.

$$P(A|B) = \frac{P(B|A).P(A)}{P(B)} \quad (1)$$

K-Nearest Neighbors (KNN): Sorts feelings into groups by looking at nearby features in large feature areas. Tested with different embeddings and n-gram combinations, focusing on choices based on similarity and the effect of semantically close reviews while showing how textual data is structured in neighborhoods.

$$distance(x, X_i) = \sqrt{\sum_{j=1}^{d} \left(x_j - X_{i_j}\right)^2} \quad (2)$$

Decision Tree: Makes splits in a hierarchy based on the best feature thresholds, using both embeddings and n-gram levels to record both frequency-based and contextual semantics. It is very good at finding nonlinear relationships and showing how opinion patterns change over structured textual features.

$$I\,(i) = 1 - \sum_{i=1}^{k} p_i^2 \quad (3)$$

Random Forest: Combines several decision trees by bagging them, which makes them more stable and useful in general. When tested with embeddings and n-grams, it does a good job of combining statistical and semantic features, which keeps it from being too good at one thing while still catching a wide range of sentiment patterns.

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

$$Gini = 1 - \sum_{i=1}^{C}(Pi)^2 \quad (4)$$

Linear Support Vector Classifier (SVC): Makes hyperplanes that separate feeling classes as much as possible in spaces with a lot of dimensions. It can work with both sparse and dense embeddings as well as n-grams, finding boundaries where features meet and ensuring strong classification across textual representations.

XGBoost: uses gradient boosting on decision trees to make classification flexible and accurate. It has been tested with different embeddings and n-grams and can handle both sparse and dense inputs. It also uses regularization to avoid overfitting and quickly pulls out fine-grained sentiment patterns.

$$\hat{y}_i = \sigma\left(\sum_{k=1}^{K} f_k(x_i)\right), f_k \in F \quad (5)$$

ElectraModel: uses transformer-based embeddings and replaced token detection to describe features efficiently and with awareness of the context. When combined with n-grams, it catches semantic subtleties and small changes in opinion, doing a better job than traditional embeddings at understanding context and telling the difference between different feelings.

h) Integration of XAI and Flask Framework:

Using Explainable AI (XAI) methods along with the Flask framework makes sentiment analysis clear and interactive. XAI techniques, like LIME and SHAP, help us understand how different words or phrases affect the predicted mood by showing us how they contribute to the whole. These methods improve trust and make it easier to understand complex embeddings and machine learning classifiers by explaining model outputs. This makes sure that forecasts are not seen as mysterious results. Users can look at how n-grams, embeddings, and hybrid feature vectors affect mood classification by finding the most important textual elements.

Flask is a simple web platform that can be used to quickly set up the sentiment analysis system. It has an interface that lets users enter text, see reasons for predictions, and interact with topic modeling results. This integration combines the ability to understand models with the ability to access them. This allows for dynamic sentiment exploration while keeping computations low and deployment easy for users.

4. EXPERIMENTAL RESULTS

Accuracy: How well a test can tell the difference between sick and healthy people is called its accuracy. To get an idea of how accurate a test is, we should figure out what percentage of cases are true positives and true negatives. In terms of math, this can be written as

$$Accuracy = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} (6)$$

Precision: Precision is the percentage of correctly classified cases or samples compared to those that were correctly classified as positives. So, here is the method to figure out the precision:

$$Precision = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} (7)$$

Recall: In machine learning, recall is a metric that shows how well a model can find all the important instances of a certain class. It shows how well a model captures instances of a certain class. It is calculated by dividing the number of correctly predicted positive observations by the total number of real positives.

$$Recall = \frac{\text{TP}}{\text{TP} + \text{FN}} (8)$$

F1-Score: F1 score is a machine learning evaluation metric that measures a model's accuracy. It combines the precision and recall scores of a model. The accuracy metric computes how many times a model made a correct prediction across the entire dataset.

$$F1\ Score = 2 * \frac{Recall\ \text{X}\ Precision}{Recall + Precision} * 100 (9)$$

Table.1 Performance Evaluation Table

Embedding	Algorithm	N-Gram	Accuracy	Precision	Recall	F1-Score
Ext BERT + BoW + TF-IDF	KNN	Unigram	0.667933	0.673677	0.668300	0.665463
Ext BERT + BoW + TF-IDF	KNN	Bigram	0.639933	0.639993	0.639875	0.639832
Ext BERT + BoW + TF-IDF	KNN	Trigram	0.643067	0.643207	0.642988	0.642898
Ext BERT	KNN	Uni +	0.667800	0.669985	0.668025	0.666911

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

+ BoW + TF-IDF		Bigram				
Ext BERT + BoW + TF-IDF	KNN	Bigram + Trigram	0.638800	0.638872	0.638738	0.638686
Ext BERT + BoW + TF-IDF	KNN	Uni + Bigram + Trigram	0.669133	0.671220	0.669352	0.668297
Ext BERT + BoW + TF-IDF	RF	Unigram	0.742267	0.742500	0.742197	0.742165
Ext BERT + BoW + TF-IDF	RF	Bigram	0.700400	0.700721	0.700309	0.700216
Ext BERT + BoW + TF-IDF	RF	Trigram	0.699400	0.699752	0.699305	0.699200
Ext BERT + BoW + TF-IDF	RF	Uni + Bigram	0.744200	0.744384	0.744138	0.744116
Ext BERT + BoW + TF-IDF	RF	Bigram + Trigram	0.710000	0.710236	0.709923	0.709866
Ext BERT + BoW + TF-IDF	RF	Uni + Bigram + Trigram	0.743200	0.743423	0.743132	0.743103
Ext BERT + BoW + TF-IDF	DT	Unigram	0.654133	0.654137	0.654139	0.654133
Ext BERT + BoW	DT	Bigram	0.606600	0.606635	0.606624	0.606595
+ TF-IDF						
Ext BERT + BoW + TF-IDF	DT	Trigram	0.589733	0.589768	0.589758	0.589728
Ext BERT + BoW + TF-IDF	DT	Uni + Bigram	0.657267	0.657260	0.657258	0.657259
Ext BERT + BoW + TF-IDF	DT	Bigram + Trigram	0.597267	0.597295	0.597288	0.597264
Ext BERT + BoW + TF-IDF	DT	Uni + Bigram + Trigram	0.658533	0.658545	0.658545	0.658533
Ext BERT + BoW + TF-IDF	SVC	Unigram	0.815333	0.815350	0.815315	0.815322
Ext BERT + BoW + TF-IDF	SVC	Bigram	0.787333	0.787352	0.787313	0.787319
Ext BERT + BoW + TF-IDF	SVC	Trigram	0.759200	0.759196	0.759197	0.759196
Ext BERT + BoW + TF-IDF	SVC	Uni + Bigram	0.818267	0.818276	0.818253	0.818258
Ext BERT + BoW + TF-IDF	SVC	Bigram + Trigram	0.785733	0.785743	0.785716	0.785722
Ext BERT + BoW + TF-IDF	SVC	Uni + Bigram+ Trigram	0.818333	0.818338	0.818322	0.818326

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

Ext BERT + BoW + TF-IDF	XGB	Unigram	0.750333	0.750333	0.750337	0.750332
Ext Electra	Electra	None	0.911200	0.912100	0.911100	0.911100

In Table 1, accuracy, precision, recall, and F1-score are used to compare how well models work across embeddings, methods, and n-grams. With its embeddings, Electra does better than all the other models and gets the best scores.

Fig.3 Comparison Graph

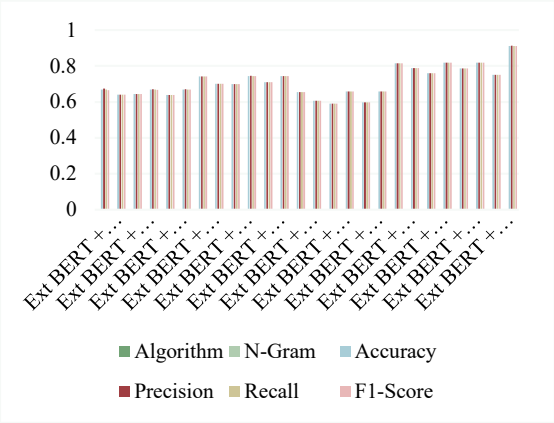

Figure 3 shows how well the model did across embeddings and algorithms. Accuracy is shown in blue, precision in red, recall in yellow, and F1-score in green, which shows that Electra had better results.

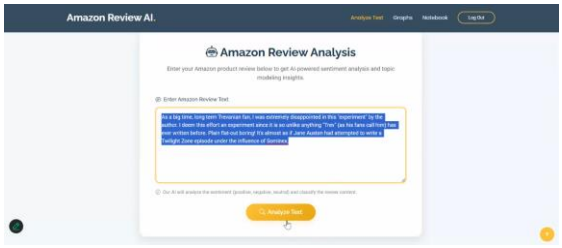

Fig.4 Enter Input Data

In Fig.4, you can see an easy-to-use interface where you can enter text to classify mood and get instant predictions.

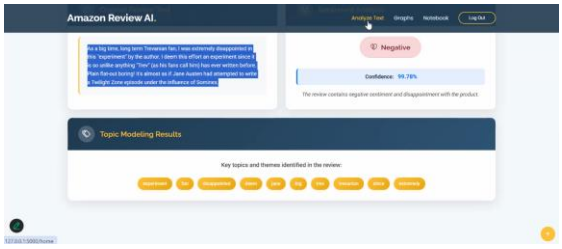

Fig.5 Output Results

The result from the sentiment classification is shown in Fig.5. The text that was entered was marked as negative with a high confidence score of 99.78%.

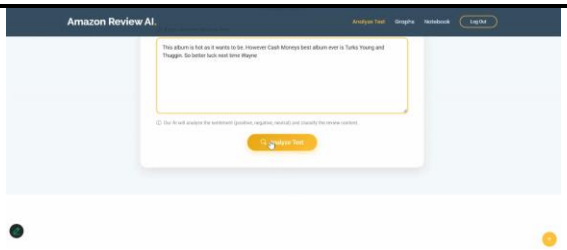

Fig.6 Enter Input Data

Figure 6 shows how to predict the sentiment of another input. The system reads the text and quickly comes up with confident classification results.

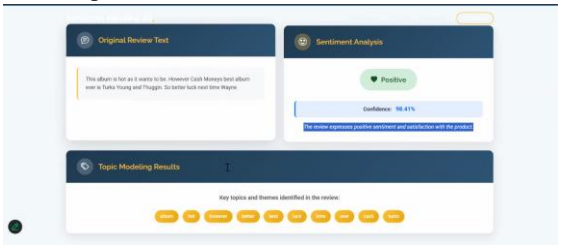

Fig.7 Output Results

The results of the mood analysis can be seen in Fig.7. With a confidence score of 98.41%, the text that was entered is marked as positive.

5. CONCLUSION

In conclusion, sentiment analysis of Amazon product reviews shows that advanced natural language processing and machine learning methods can effectively pull out opinions and feelings from written data. Using a hand-picked subset of 50,000 reviews from the original 4,000,000-record Kaggle dataset, preprocessing steps like normalization, HTML and URL removal, tokenization, label encoding, and n-gram creation made it possible to extract structured features and make the model more ready. Several embedding methods were used to create feature vectors, such as Bag-of-Words (BoW), TF-IDF, Word2Vec, BERT, and a combination of BERT and TF-IDF. These vectors gave rich semantic and statistical representations of textual material. We learned and tested machine learning classifiers like Naive Bayes, K-Nearest Neighbors, Decision Tree, Random Forest, and Linear SVC using accuracy, precision, recall, F1-score, and confusion matrices to make sure they worked well. To improve the accuracy and readability of the predictions, advanced models like XGBoost and ElectraModel were combined with hybrid embeddings that combined BERT, TF-IDF, and BoW. Explainable AI methods like LIME and SHAP were also used to show which features were most important. Electra embeddings with

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

ElectraModel worked better than baseline and conventional embeddings, achieving 91.1% accuracy. This shows that hybrid embeddings and advanced classifiers are useful for classifying sentiment.

In the future, this system could be used to look into deeper transformer-based designs like RoBERTa, XLNet, and GPT variants to make sentiment classification even better. Adding reviews written in more than one language to the dataset would make it more useful on more global sites. Streaming mood analysis from social media and e-commerce sites in real time can give you dynamic information about what customers think. Adding recommendation systems to the mix could make personalized product ideas better. Also, more advanced explainable AI methods besides LIME can make it easier to understand, which ensures trustworthiness and transparency. With scalable microservices and cloud-based deployment, huge datasets can be handled quickly, and the system can be used for large-scale industry tasks.

REFERENCES

[1] Qorich, M., & El Ouazzani, R. (2023). Text sentiment classification of Amazon reviews using word embeddings and convolutional neural networks. The Journal of Supercomputing, 79(10), 11029-11054.

[2] Ouni, C., Benmohamed, E., & Ltifi, H. (2024). Sentiment analysis deep learning model based on a novel hybrid embedding method. Social Network Analysis and Mining, 14(1), 210.

[3] Erkan, A., & Güngör, T. (2023). Analysis of Deep Learning Model Combinations and Tokenization Approaches in Sentiment Classification. IEEE Access, 11, 134951-134968.

[4] Sabbeh, S. F., & Fasihuddin, H. A. (2023). A Comparative analysis of word embedding and deep learning for Arabic sentiment classification. Electronics, 12(6), 1425.

[5] Yafoz, A., & Mouhoub, M. (2021, October). Sentiment analysis in Arabic social media using deep learning models. In 2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC) (pp. 1855-1860). IEEE.

[6] G, Viswanath., N, Madhvik., K, Bhaskar., K, Supriya. (2024). Machine-Learning-Based Cloud Intrusion Detection. International Journal of Mechanical Engineering Research and Technology, 16(9), 38–52.

[7] A. Ullah, S. N. Khan, and N. M. Nawi, "Review on sentiment analysis for text classification techniques from 2010 to 2021," Multimedia Tools Appl., vol. 82, no. 6, pp. 8137–8193, Mar. 2023.

[8] Z. Lin, J. Tan, D. Ou, X. Chen, S. Yao, and B. Zheng, "Deep bag-of-words model: An efficient and interpretable relevance architecture for Chinese e-commerce," in Proc. 30th ACM SIGKDD Conf. Knowl. Discovery Data Mining, Aug. 2024, pp. 5398–5408.

[9] J.-W. Sun, J.-Q. Bao, and L.-P. Bu, "Text classification algorithm based on TF-IDF and BERT," in Proc. 11th Int. Conf. Inf. Commun. Technol. (ICTech), Feb. 2022, pp. 1–4.

[10] R. H. AlMahmoud and B. H. Hammo, "SEWAR: A corpus-based N-gram approach for extracting semantically-related words from Arabic medical corpus," Expert Syst. Appl., vol. 238, Mar. 2024, Art. no. 121767.

[11] K. Kobra, S. S. Sammi, N. Rahman, S. A. Khushbu, and M. Islam, "Multihead text mining from COVID-19 feedback using machine learning, deep learning, and hybrid deep learning approaches," J. Sensors, vol. 2024, no. 1, Jan. 2024, Art. no. 3027199.

[12] C. Yang and C. Ding, "Learning word embedding with better distance weighting and window size scheduling," 2024, arXiv:2404.14631.

[13] J. M.D. Rezende, I. M. D.C. Rodrigues, L. C. Resendo, and K. S. Komati, "Combining natural language processing techniques and algorithms LSA, word2Vec and WMD for technological forecasting and similarity analysis in patent documents," Technol. Anal. Strategic Manage., vol. 36, no. 8, pp. 1695–1716, Aug. 2024.

[14] A. Qadir Mir, F. Yaqub Khan, and M. Ahsan Chishti, "Online fake review detection using supervised machine learning and BERT model," 2023, arXiv:2301.03225.

[15] S. González-Carvajal and E. C. Garrido-Merchán, "Comparing BERT against traditional machine learning text classification," 2020, arXiv:2005.13012.

[16] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, arXiv:1810.04805.

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

[17] H. Zou and Z. Wang, "A semi-supervised short text sentiment classification method based on improved BERT model from unlabelled data," J. Big Data, vol. 10, no. 1, pp. 1–19, Mar. 2023.
[18] R. Horev. "BERT Explained: State of the Art Language Model for NLP—Towardsdatascience.com." Accessed: Oct. 28, 2024. [Online]. Available: https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270
[19] Sowmya, G., & Swapna, G. (2023). A Real Time Online Food Ordering Application Based Django Restful Framework. Industrial Engineering Journal, 52(9), 425–431.
[20] M. Belguith, C. Aloulou, and B. Gargouri, "Aspect-level sentiment analysis based on deep learning and ontologies," Social Netw. Comput. Sci., vol. 5, no. 1, p. 58, Dec. 2023.
[21] S. Gharatkar, A. Ingle, T. Naik, and A. Save, "Review preprocessing using data cleaning and stemming technique," in Proc. Int. Conf. Innov. Inf., Embedded Commun. Syst. (ICIIECS), Mar. 2017, pp. 1–4.
[22] S. L. Mahfiz and A. Romadhony, "Aspect-based opinion mining on beauty product reviews," in Proc. 3rd Int. Seminar Res. Inf. Technol. Intell. Syst. (ISRITI), Dec. 2020, pp. 488–493.
[23] G Loge, T Sunil Kumar Reddy, G Swapna, & G Viswanath. (2025). Interpretable AI for Precision Brain Tumor Prognosis: A Transparent Machine Learning Approach. In International Journal of Health Sciences and Pharmacy (IJHSP) (Vol. 9, Number 1, pp. 180–195). Zenodo. https://doi.org/10.5281/zenodo.15523628
[24] N. J. Euna, S. M. M. Hossain, M. M. Anwar, and I. H. Sarker, "Content based spam email detection using an n-gram machine learning approach," in Applied Intelligence for Industry 4.0, London, U.K.: Chapman & Hall, 2023, pp. 176–187.
[25] G. Angiani, L. Ferrari, T. Fontanini, P. Fornacciari, E. Iotti, F. Magliani, and S. Manicardi, "A comparison between preprocessing techniques for sentiment analysis in Twitter," in Proc. KDWeb, 2016, pp. 1–11.
[26] P. Yadav and D. Pandya, "SentiReview: Sentiment analysis based on text and emoticons," in Proc. Int. Conf. Innov. Mech. Ind. Appl. (ICIMIA), Feb. 2017, pp. 467–472.
[27] D. E. Cahyani and I. Patasik, "Performance comparison of TF-IDF and Word2Vec models for emotion text classification," Bull. Electr. Eng. Informat., vol. 10, no. 5, pp. 2780–2788, Oct. 2021.
[28] T. E. Trueman, A. K. Jayaraman, E. Cambria, G. Ananthakrishnan, and S. Mitra, "An N-gram-based BERT model for sentiment classification using movie reviews," in Proc. Int. Conf. Artif. Intell. Data Eng. (AIDE), Dec. 2022, pp. 41–46.
[29] Kumar, C. S., Sirisati, R. S., Gudditti, V., Rao, K. S., & Challa, R. K. (2022, December). A smart recommendation system for medicine using intelligent NLP techniques. In 2022 International Conference on Automation, Computing and Renewable Systems (ICACRS) (pp. 1081–1084). IEEE. https://doi.org/10.1109/ICACRS55517.2022.10029078
[30] A. E. Qasem and M. Sajid, "Exploring the effect of N-grams with BOW and TF-IDF representations on detecting fake news," in Proc. Int. Conf. Data Anal. Bus. Ind. (ICDABI), Oct. 2022, pp. 741–746.