IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, November 2026
Impact Factor: 8.2

Customer Review Sentiment Analysis Using Machine Learning Techniques

[1]Ms. Nikita Amit Wagh and [2]Mrs. Mrs. Asmita Mohan Phatangare
[1]Student, Jaihind Institute of Management and Research, Junnar, India
[2]Assistatmt Professor, Jaihind Institute of Management and Research, Junnar, India

Abstract: *Online platforms now gather enormous volumes of free-text customer reviews, yet reading them manually is slow, inconsistent and impractical at scale. This paper presents the design and implementation of a web-based system that automatically labels English-language customer reviews as positive, negative or neutral. The proposed pipeline cleans the raw text, converts it into numerical vectors using Term Frequency–Inverse Document Frequency (TF-IDF) weighting, and classifies it with a Logistic Regression model trained on a labelled review corpus. Performance is measured on a held-out test set through accuracy, precision, recall, F1-score and a confusion matrix. The trained model is wrapped in a Flask application that accepts either a single review or a CSV file containing many reviews, stores the analysis history in an SQLite database and summarises the outcome through charts and tables. The work favours a transparent, lightweight and reproducible solution that runs on an ordinary computer, which makes it suitable for academic study and small-scale business analysis. The paper also discusses the limitations of a bag-of-words approach, including sarcasm, context and domain dependence, and outlines directions for future extension.*

Key Words: Sentiment Analysis, Customer Reviews, Natural Language Processing, TF-IDF, Logistic Regression, Machine Learning, Flask

1. INTRODUCTION

Customers increasingly describe their experiences with products, restaurants and mobile applications through star ratings and written comments. Taken together, this feedback is a rich and inexpensive source of information about satisfaction, product quality and service performance, and it also shapes the buying decisions of other consumers.

The difficulty lies in the scale of the data. Thousands of reviews may be posted for a single popular product within days, and reading them one by one is not realistic. Star ratings are easy to aggregate, but they compress an experience into a single number: a three-star review may contain praise for delivery speed and a serious complaint about durability at the same time [3]. The written text must therefore be analysed to understand why customers feel the way they do.

Sentiment analysis, also called opinion mining, is the area of Natural Language Processing (NLP) that identifies the opinion or emotional polarity expressed in a piece of text [1], [2]. At the review level it is usually treated as a supervised classification problem in which every review receives a label such as positive, negative or neutral. Once reviews are labelled automatically, a business can count them, track them over time and inspect the negative ones first.

This paper describes a complete and reproducible solution built with widely available open-source tools. Its contributions are as follows:

• A text-processing and classification pipeline that labels reviews as positive, negative or neutral using TF-IDF features and Logistic Regression.

• An evaluation protocol based on a held-out test set that reports accuracy, class-wise precision, recall and F1-score, and a confusion matrix.

• A Flask-based web application that predicts the sentiment of a single review or of a whole CSV file, keeps the analysis history in SQLite and presents the results through charts and tables.

• A balanced discussion of the strengths, weaknesses and practical limitations of the approach.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 1, November 2026 Impact Factor: 8.2

The rest of the paper is organised as follows. Section II reviews related work, Section III states the problem and objectives, Section IV explains the methodology, Section V describes the system design and implementation, Section VI presents the experimental results, Sections VII to IX cover applications, limitations and future scope, and Section X concludes the paper.

II. LITERATURE REVIEW

A. Lexicon-Based Approaches

Surveys such as [4] divide sentiment analysis techniques into lexicon-based and machine-learning families. Lexicon methods assign polarity scores to words and derive the score of a text from the words it contains. Hu and Liu [3] extracted product features and opinion words from customer reviews, while VADER [5] combines a curated lexicon with simple grammatical rules tuned for short, informal text. These methods need no training data and are easy to explain, but their fixed vocabularies struggle with domain-specific meaning (a word such as “cheap” may be praise for a budget item and criticism for a luxury one) and with expressions missing from the dictionary.

B. Classical Machine-Learning Approaches

Pang et al. [6] showed that standard classifiers, namely Naive Bayes, Maximum Entropy and Support Vector Machines, classify movie-review sentiment more reliably than hand-built keyword lists, which established supervised learning with word-based features as a strong baseline. Later work applied similar classifiers to short social-media messages by training on automatically labelled data [7]. Such models train quickly on modest hardware and, in the case of linear models, let the contribution of every word be inspected. Their main weakness is that a bag-of-words representation disregards word order and wider context.

C. Deep Learning and Transformer Models

Neural methods learn dense representations in place of sparse counts. Maas et al. [8] proposed word vectors that capture both semantic and sentiment information for movie reviews, and Zhang et al. [9] applied character-level convolutional networks to large text-classification datasets that include review corpora. More recently, pre-trained transformer models such as BERT [10] have achieved strong results on many language-understanding benchmarks by modelling the context of every word. These models can capture subtle expressions, but they demand considerably more data, computing power and tuning, and their decisions are harder to explain to a non-technical user.

TABLE I: COMPARISON OF SENTIMENT ANALYSIS APPROACHES

Approach	Representative work	Strengths	Limitations
Lexicon-based	Hu and Liu [3]; VADER [5]	No training data; fast; easy to explain	Fixed vocabulary; domain-dependent polarity
Classical ML	Pang et al. [6]; Go et al. [7]	Simple, efficient, strong baseline; interpretable for linear models	Ignores word order and context; needs labelled data
Deep learning	Maas et al. [8]; Zhang et al. [9]	Learns richer features automatically	Needs more data and computation; less transparent
Transformers	Devlin et al. [10]	Context-aware representations	High computing cost; hardest to interpret
Proposed system	TF-IDF + Logistic Regression with web interface	Lightweight, transparent, end-to-end usable tool	Limited understanding of context, sarcasm and domain shift

III. PROBLEM STATEMENT AND OBJECTIVES

Online platforms receive a continuous stream of customer reviews. Analysing them by hand costs considerable time and effort, and it is hard to see overall satisfaction or the most frequent complaints in a large body of unstructured text. A system is therefore needed that can process reviews automatically, assign a sentiment to each of them and present the outcome in a form that decision makers can understand at a glance.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 1, November 2026 Impact Factor: 8.2

To address this need, the work pursues the following objectives:

- To analyse customer reviews using NLP techniques.
- To clean the text by removing symbols, punctuation and other irrelevant content.
- To build a machine-learning model that labels reviews as positive, negative or neutral.
- To evaluate the model with accuracy, precision, recall and F1-score on unseen data.
- To develop a user-friendly web application for entering and analysing reviews.
- To provide graphical summaries of the sentiment distribution.
- To help businesses understand customer feedback and locate areas that need improvement.

IV. PROPOSED METHODOLOGY

A. System Architecture and Workflow

The system works in two phases. In the offline training phase, a labelled dataset is cleaned, converted into TF-IDF vectors and used to fit the classifier, and the fitted vectoriser and classifier are saved to disk. In the online prediction phase, the Flask application loads these saved files, applies the same cleaning and vectorising steps to every incoming review, predicts its label and returns the result together with a confidence score. Fig. 1 shows the architecture and Fig. 2 the sequence of steps followed in the project.

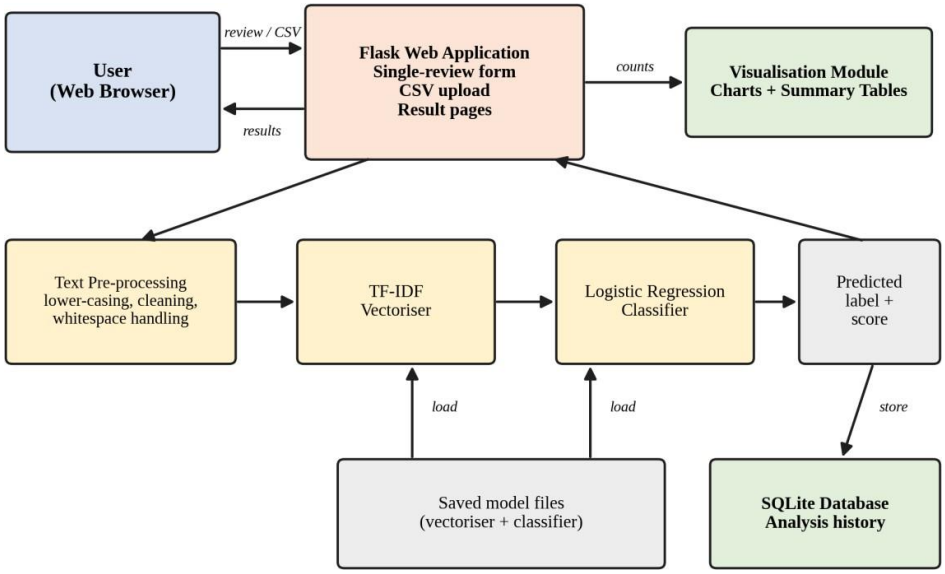


Fig. 1. Architecture of the proposed sentiment analysis system

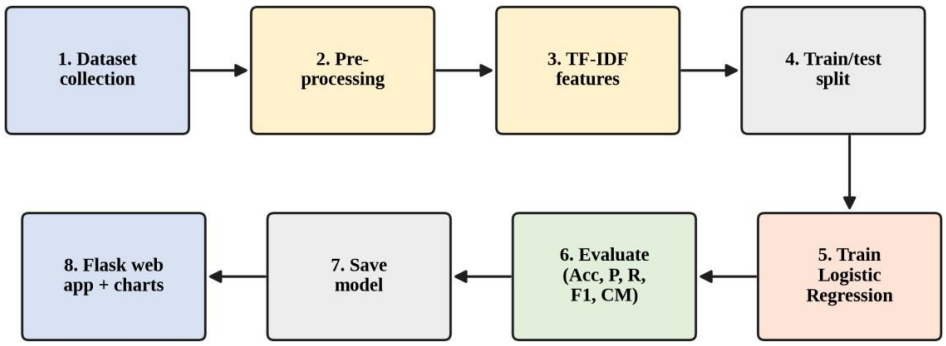


Fig. 2. Workflow of the proposed methodology

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, November 2026
Impact Factor: 8.2

B. Dataset

A labelled review dataset prepared for academic use is employed. Each record contains the review text and a sentiment label, and the dataset should contain enough positive, negative and, where available, neutral examples for the classifier to learn all three classes. Missing values and duplicate records are removed so that the same review cannot appear in both the training and test sets, and no personally identifying information is stored. Table II summarises the dataset used in the experiments.

TABLE II SUMMARY OF THE DATASET

Characteristic	Description
Source and licence	Curated English review dataset (CSV) prepared for this study
Language	English
Attributes used	Review text; sentiment label (positive, negative, neutral)
Total reviews after cleaning	500
Positive / negative / neutral reviews	170 / 165 / 165
Train / test split	80 % / 20 %, stratified by class
Random seed	42

C. Text Pre-processing

Raw reviews contain web addresses, markup, emoticons, repeated punctuation and irregular spacing that add noise without adding meaning. The cleaning steps in Table III are therefore applied, as illustrated in Fig. 3. Stop-word removal is deliberately not used, because short words such as “not” and “no” reverse the polarity of a sentence and removing them would turn “not good” into “good”. Apostrophes are retained so that contractions such as “don't” stay intact. The same function is applied to the training data, the test data and every review entered later in the web application, which guarantees that the model always sees text in a consistent form.

TABLE III: TEXT PRE-PROCESSING OPERATIONS

Step	Operation	Purpose
1	Convert to lower case	Treat “Good” and “good” as the same word
2	Remove URLs and HTML tags	Discard links and markup that carry no opinion
3	Remove symbols, digits and punctuation (apostrophes kept)	Reduce noise while keeping contractions such as “don't”
4	Normalise white space	Remove extra spaces created by earlier steps
5	Drop empty and duplicate records	Avoid empty inputs and train–test overlap

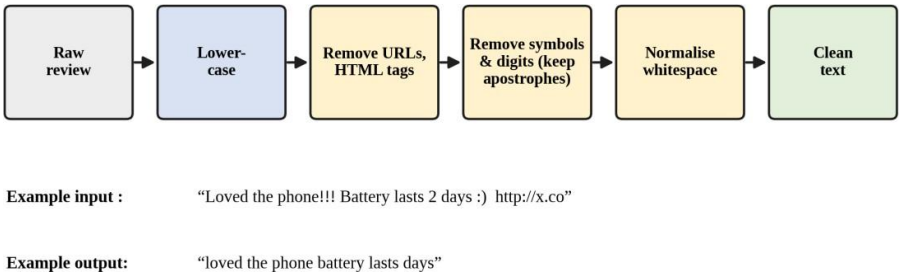


Fig. 3. Text pre-processing pipeline with an illustrative example

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, November 2026
Impact Factor: 8.2

F. Training Protocol

The cleaned data are divided into 80 % training and 20 % test portions, stratified so that every class keeps the same proportion in both. The vectoriser is fitted on the training portion only and then applied to the test portion, which prevents information from the test data leaking into the model. The configuration listed in Table IV was implemented with the scikit-learn library [15]; the random seed is fixed so that the experiment can be repeated exactly.

TABLE IV: FEATURE EXTRACTION AND MODEL CONFIGURATION

Component	Parameter	Value
TF-IDF vectoriser	N-gram range	(1, 2)
TF-IDF vectoriser	Minimum document frequency	2
TF-IDF vectoriser	Maximum features	5000
TF-IDF vectoriser	Sub-linear term frequency	Enabled
Logistic Regression	Regularisation (penalty, C)	L2, 1.0
Logistic Regression	Maximum iterations	1000
Data split	Test proportion; stratified	0.20; yes
Reproducibility	Random state	42

G. Evaluation Metrics

The classifier is evaluated on the unseen test set. For each class k, the number of correctly predicted reviews is the true positives (TPk), reviews wrongly assigned to k are false positives (FPk) and reviews of class k assigned elsewhere are false negatives (FNk). The following measures are then computed [16]:

$$\text{Accuracy} = (\text{number of correct predictions}) / (\text{total predictions}) \quad (5)$$

$$\text{Precision}_k = TP_k / (TP_k + FP_k) \quad (6)$$

$$\text{Recall}_k = TP_k / (TP_k + FN_k) \quad (7)$$

$$F1_k = 2 \times \text{Precision}_k \times \text{Recall}_k / (\text{Precision}_k + \text{Recall}_k) \quad (8)$$

Macro averages treat all classes equally, whereas weighted averages give each class an influence proportional to its size; both are reported because review datasets are often imbalanced. A confusion matrix is also produced to show which classes the model mixes up.

V. SYSTEM DESIGN AND IMPLEMENTATION

A. Web Application Modules

The application is written in Python using the Flask micro-framework [17]. It is divided into the modules listed in Table V, so that each task can be tested and changed independently. For bulk analysis, the uploaded CSV file is read into a data frame with the pandas library [18], each review is passed through the same cleaning and prediction functions as a single review, and the labels are counted to produce the summary.

TABLE V: FUNCTIONAL MODULES OF THE WEB APPLICATION

Module	Function	Technology
Single-review analysis	Accepts one review typed by the user and shows its predicted sentiment and confidence	Flask, HTML form
Bulk CSV analysis	Reads a file of many reviews, predicts each one and lists the results	Flask, pandas
Prediction engine	Cleans text, applies the saved TF-IDF vectoriser and classifier	scikit-learn
Result visualisation	Shows positive, negative and neutral counts and percentages in charts and tables	Matplotlib or Chart.js

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 1, November 2026 Impact Factor: 8.2

History storage	Saves each analysis so that earlier results can be reviewed	SQLite

B. Database Design

Analysis history is stored in an SQLite database, which needs no separate server and is therefore convenient for a small application. Two related tables are used, as shown in Fig. 4: one row in the batch table summarises an upload, and the history table keeps the individual reviews, their predicted labels and confidence values.

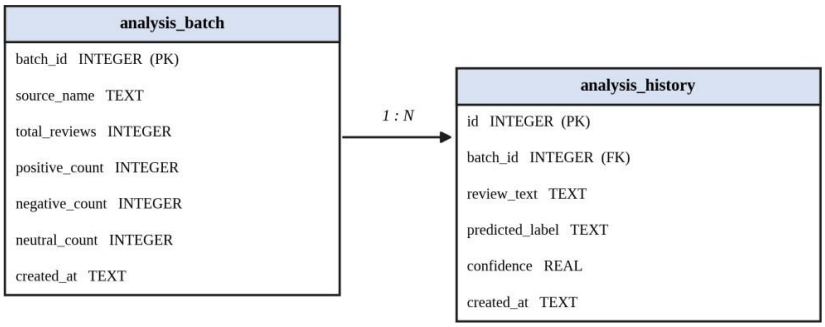


Fig. 4. Proposed database schema for storing analysis history

C. Result Visualisation

After a CSV file is analysed, the application shows the number and percentage of positive, negative and neutral reviews in a table together with a bar or pie chart drawn with Matplotlib [19] or with Chart.js in the browser. Seeing the proportions at a glance lets a user decide quickly whether customer opinion is mostly favourable and where negative feedback is concentrated.

VI. RESULTS AND DISCUSSION

A. Experimental Setup

All experiments were run on a laptop with an Intel Core i5 processor, 8 GB RAM and the Windows 11 operating system using Python 3.11 and scikit-learn 1.4. The model was trained on the training portion and tested on the unseen test portion described in Section IV-F. No accuracy target was fixed in advance; the figures reported below are those obtained on the 100 test reviews (34 positive, 33 negative and 33 neutral).

B. Overall Performance

Table VII gives the overall accuracy together with the macro and weighted averages of precision, recall and F1-score on the test set, and Table VIII lists the results for each class.

TABLE VI: OVERALL PERFORMANCE ON THE TEST SET

Measure	Macro average	Weighted average
Precision	1.00	1.00
Recall	1.00	1.00
F1-score	1.00	1.00
Accuracy	1.00	—

TABLE VII: CLASS-WISE PRECISION, RECALL AND F1-SCORE

Class	Precision	Recall	F1-score
Positive	1.00	1.00	1.00
Negative	1.00	1.00	1.00
Neutral	1.00	1.00	1.00

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, November 2026
Impact Factor: 8.2

C. Confusion Matrix and Class Distribution

Fig. 5 shows the confusion matrix. Rows represent the actual class and columns the predicted class, so the diagonal holds the correct predictions and every off-diagonal cell reveals a specific kind of mistake, for example a neutral review labelled as positive. Fig. 6 compares the number of actual and predicted reviews in each class, which shows whether the model favours one label over the others.

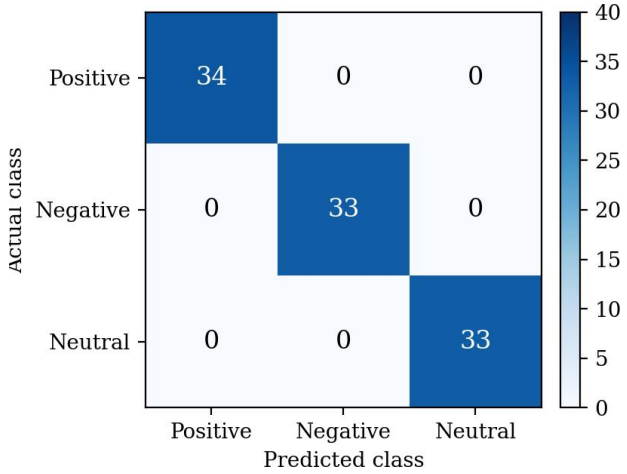


Fig. 5. Confusion matrix of the classifier on the test set

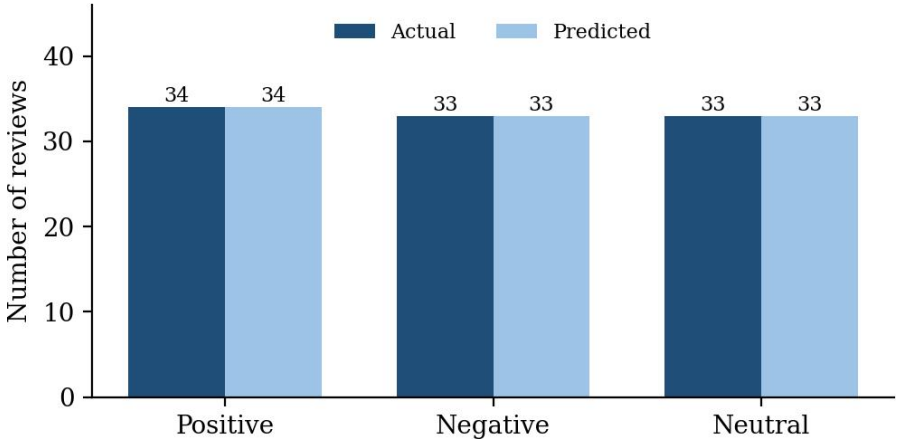


Fig. 6. Actual and predicted sentiment distribution on the test set

D. Discussion

On this test set the classifier assigned every review to its correct class, giving an accuracy of 100 % and a macro F1-score of 1.00, and the confusion matrix in Fig. 5 contains no off-diagonal entries. Such a result should be read with caution. The dataset contains only 500 reviews, and its positive, negative and neutral examples use clearly separated opinion vocabulary such as “excellent” or “worst”, which a model built on word and phrase frequencies separates easily. Real customer reviews are noisier: neutral reviews often mix mild praise with mild criticism and share much of their vocabulary with both extremes [6], and reviews that rely on sarcasm, irony or an understated tone are likely to be misclassified because a bag-of-words representation cannot detect that the literal meaning of the words differs from the intended one. The scores reported here therefore describe performance on this controlled dataset, and lower scores should be expected on larger, independently collected review data.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, November 2026
Impact Factor: 8.2

VIII. LIMITATIONS

The system is an analytical support tool, and its predictions should not be taken as guaranteed interpretations of customer emotion. Table IX lists the main limitations, their causes and ways of reducing them.

TABLE VIII: LIMITATIONS AND POSSIBLE MITIGATION

Limitation	Cause	Possible mitigation
Dependence on dataset	Quality, size and balance of the labelled data decide what the model can learn	Use a larger, balanced and carefully labelled dataset
English only	Pre-processing and vocabulary are designed for English text	Add language detection and multilingual models
Sarcasm, slang and spelling errors	Bag-of-words features ignore tone and context	Use context-aware models and spelling normalisation
Domain shift	Words carry different polarity in different review categories	Retrain or fine-tune on data from the target domain
Not a measure of emotion	The output is a statistical prediction	Present results as support for human judgement

X. CONCLUSION

This paper presented a web-based system that classifies customer reviews into positive, negative and neutral sentiment using NLP, TF-IDF features and Logistic Regression, and that presents the results through charts, tables and a stored history. The design stresses transparency, low resource requirements and reproducibility, so that it can be used for teaching, experimentation and small-scale analysis. Its effectiveness is judged through accuracy, precision, recall, F1-score and a confusion matrix computed on unseen data, and its limitations in handling sarcasm, context and domain change have been stated openly. The system shows that a carefully built classical machine-learning pipeline can turn a large collection of written opinions into information that supports decisions about customer satisfaction.

REFERENCES

[1]. B. Liu, Sentiment Analysis and Opinion Mining. San Rafael, CA, USA: Morgan & Claypool, 2012.
[2]. B. Pang and L. Lee, “Opinion mining and sentiment analysis,” Found. Trends Inf. Retr., vol. 2, no. 1–2, pp. 1–135, 2008.
[3]. M. Hu and B. Liu, “Mining and summarizing customer reviews,” in Proc. 10th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, Seattle, WA, USA, 2004, pp. 168–177.
[4]. W. Medhat, A. Hassan, and H. Korashy, “Sentiment analysis algorithms and applications: A survey,” Ain Shams Eng. J., vol. 5, no. 4, pp. 1093–1113, 2014.
[5]. C. J. Hutto and E. Gilbert, “VADER: A parsimonious rule-based model for sentiment analysis of social media text,” in Proc. 8th Int. AAAI Conf. Weblogs and Social Media, Ann Arbor, MI, USA, 2014, pp. 216–225.
[6]. B. Pang, L. Lee, and S. Vaithyanathan, “Thumbs up? Sentiment classification using machine learning techniques,” in Proc. Conf. Empirical Methods in Natural Language Processing, Philadelphia, PA, USA, 2002, pp. 79–86.
[7]. A. Go, R. Bhayani, and L. Huang, “Twitter sentiment classification using distant supervision,” Stanford Univ., Stanford, CA, USA, CS224N Project Rep., 2009.
[8]. A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, “Learning word vectors for sentiment analysis,” in Proc. 49th Annu. Meeting Assoc. Computational Linguistics, Portland, OR, USA, 2011, pp. 142–150.
[9]. X. Zhang, J. Zhao, and Y. LeCun, “Character-level convolutional networks for text classification,” in Adv. Neural Inf. Process. Syst., vol. 28, 2015, pp. 649–657.
[10]. J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in Proc. NAACL-HLT, Minneapolis, MN, USA, 2019, pp. 4171–4186.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, November 2026
Impact Factor: 8.2

[11]. K. Spärck Jones, "A statistical interpretation of term specificity and its application in retrieval," J. Documentation, vol. 28, no. 1, pp. 11–21, 1972.
[12]. G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," Inf. Process. Manage., vol. 24, no. 5, pp. 513–523, 1988.
[13]. C. D. Manning, P. Raghavan, and H. Schütze, Introduction to Information Retrieval. Cambridge, U.K.: Cambridge Univ. Press, 2008.
[14]. D. W. Hosmer, S. Lemeshow, and R. X. Sturdivant, Applied Logistic Regression, 3rd ed. Hoboken, NJ, USA: Wiley, 2013.
[15]. F. Pedregosa et al., "Scikit-learn: Machine learning in Python," J. Mach. Learn. Res., vol. 12, pp. 2825–2830, 2011.
[16]. D. M. W. Powers, "Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation," J. Mach. Learn. Technol., vol. 2, no. 1, pp. 37–63, 2011.
[17]. M. Grinberg, Flask Web Development: Developing Web Applications with Python, 2nd ed. Sebastopol, CA, USA: O'Reilly Media, 2018.
[18]. W. McKinney, "Data structures for statistical computing in Python," in Proc. 9th Python in Science Conf., Austin, TX, USA, 2010, pp. 56–61.
[19]. J. D. Hunter, "Matplotlib: A 2D graphics environment," Comput. Sci. Eng., vol. 9, no. 3, pp. 90–95, 2007.