iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** IV **Month of publication:** April 2026

DOI: https://doi.org/10.22214/ijraset.2026.79901

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

Intelligent Sentiment Analysis of Customer Reviews Using NLP

Prof. S V Raut [1], Mr. Shiva Wanjalkar[2], Mr. Gaurav Janbandhu[3], Mr. Sayed Aadib[4], Ms. Divya Belekar[5], Ms.Gauri Dayrekar[6]

[1]*Assistant Professor,* [2,3,4,5,6]*Undergraduate Student, Department of Computer Science & Engineering, Dr. Rajendra Gode Institute of Technology and Research, Amravati (MH), India*

Abstract: ***The current explosion of e-commerce activity has created a massive data bottleneck where thousands of daily customer reviews are generated, making manual oversight physically impossible for businesses. This Study addresses the challenge by developing an automated sentiment analysis pipeline specifically designed to process and categorize unstructured consumer feedback into positive, negative, or neutral sentiments. Utilizing a substantial corpus of over 200,000 product reviews, the system employs a rigorous Natural Language Processing (NLP) workflow that prioritizes lemmatization over basic stemming to preserve the semantic integrity of the text. By transforming the cleaned data through TF-IDF vectorization incorporating both unigrams and bigrams the model is trained to capture critical contextual nuances, such as negation, which simpler frequency-based models often misinterpret. For the classification stage, the Multinomial Naive Bayes algorithm was selected due to its high computational efficiency and proven performance with high-dimensional textual datasets. To bridge the gap between theoretical modeling and real-world application, the final system was integrated into a Flask-based web interface, providing an accessible platform for real-time sentiment prediction. Our experimental results, based on a 20% unseen test set, confirm that the model effectively distinguishes between polar sentiments while highlighting the inherent linguistic difficulty of classifying neutral feedback. Ultimately, this project provides a scalable, low-cost framework for small-to-medium enterprises to automate their customer feedback loops without requiring expensive infrastructure***

I. INTRODUCTION

Every day, more folks choose buying stuff through the internet, shifting how purchases happen. With sites pulling together countless vendors, comparing prices, details, or feedback becomes part of browsing - no extra effort needed. For many deciding which item to grab, reviews left by earlier customers carry serious weight. Surprisingly honest thoughts from real users often steer the choice more than polished ads ever could. Real talk and ratings show how things actually perform - whether they're simple to handle or leave folks satisfied afterward - not flashy promotions. Handling massive e-commerce platforms though? It brings waves of reviews pouring in daily. With so many words flooding in, reading every note one by one eats up hours. So machines quietly take over, piecing together what shoppers truly experience without needing a human eye on each line. Words carry feelings. Computers learn these tones using methods tied to understanding speech. One such method checks if written lines show happiness, anger, or quiet disappointment. Sorting happens through systems trained on patterns in data. Sometimes it uses math models named after probability rules. Other times, complex networks mimic brain-like decisions. Among them, one version works quickly on word counts transformed into digits. It thrives where phrases become number rows via specific formulas. A system built here examines comments from buyers online. Its path shaped by layers of code that notice context, weight terms, adjust guesses. Each note judged not just by what's said but how it sits among similar ones. Tossing over two hundred thousand reviews into the mix demands serious cleanup, reworking raw text into structured data through TF-IDF so word importance stands clear. Out pops a modest site made with Flask - no frills, merely accepts user input and returns an emotional label. Emotion detection takes center stage here, guided by Multinomial Naive Bayes after the text is converted into numeric form. The goal? Construct something useful: a system that reads between lines in customer feedback. Rather than boxing opinions as positive or negative, it probes deeper - the feelings hidden beneath phrases typed on shopping sites. Learning happens slowly, fed by mountains of old remarks until the algorithm spots subtle trends. What emerges isn't magic it's pattern recognition trained on real human expressions. Most of the time, learning happens through patterns found in live cases instead of strict guidelines. While one piece works on brief expressions, something else tracks how mood changes in extended passages. Meaning comes out of more than just descriptive words - sentence flow plus surrounding hints play a role too.

When fresh slang appears in customer feedback, the approach bends without breaking. What results is a sharper view of actual buyer opinions. Looking at data closely helps spot emotional trends along with basic traits of the collection.

Cleaning text begins by breaking it down with NLP tools. From there, words shift into figures via TF-IDF. That approach balances appearance count with overall rarity between files. Values reflect weight, shaped by relevance more than repetition alone. Scoring shifts per term, driven by where and how it shows up. What stands out in the text gets more attention through weighting. For training a Multinomial Naive Bayes model to predict sentiment, start here. Numbers tell part of the story - accuracy shows one angle. Yet precision brings another layer when viewed beside it. Because recall plays a role, ignoring it skews understanding. Since the F1-score balances both, it rounds out the view. Depending on the metric, importance shifts subtly across outcomes. Though each focuses elsewhere, together they map performance piece by piece. Side by side, they form a full picture without depending on any single piece. Built using Flask, the web application accepts entries from users. After that, it judges whether the text carries a bright or dark tone. Rather than gathering feedback silently, it answers with a mood signal. Words go in, emotions come out. Something whispers a hunch about feeling. Words spill out, shaped into guesses about stance. From plain stuff, fast judgments rise on how things sound. A thing grows here - meant to sort what buyers say online, split by emotional color. Skip the heavy math machines; no LSTM, no Transformer circus. Choose quiet logic instead: Naïve Bayes hums, calling views positive, negative, or blank. Even with more than two hundred thousand records listing things such as titles, prices, ratings, headlines, and short descriptions, just the English comments move through the system. Instead of using neural networks like most current tools, it relies on simpler, time-tested methods. After each message is checked, it lands in one of three emotion categories. Although limited to a single language, the setup manages vast amounts of user feedback. Older techniques aren't outdated - sometimes they're exactly what the job needs. Out of jumbled words, emotions split into positive, negative, or neutral ones. Cleaning happens before anything else - removing clutter that clouds meaning. Then comes turning phrases into digits using something known as TF-IDF. After that, a system labeled Multinomial Naive Bayes uses those figures to predict mood. One piece leads to another, keeping things running like gears in sequence. Each round finishes with exactly three results. One step follows another, yet it seems lightning fast. Nothing gets marked unless it fits a known shape. Words shift again and again until the last cluster forms. Phrases turn into just one of those three signs in the end.

A. Aim

This project aims to create an automated system capable of organizing shopper feedback from e-commerce sites - tagging each thought as positive, negative, or neutral through machine-driven language analysis. With online shopping rising, digital storefronts face massive waves of reviews every day, far beyond what teams can check manually. As software detects word trends, personal monitoring slips away under algorithmic handling. The moment a remark enters, it flows straight into evaluation, never waiting in line. Some meanings hide behind quiet words; spotting them means looking closer. Yet hints appear when you watch how phrases repeat themselves. Comments that seem flat might carry weight if heard again. Learning happens by comparing today's messages to yesterday's lessons. With more examples, even twisted phrasings start making sense. Over months without fresh tweaks, systems slowly stumble on new slang twists. Hidden cycles of user replies nudge accuracy forward, bit by bit. Here's how it works. When processing large amounts of written feedback, speed stays steady thanks to an efficient structure. Texts get split into pieces right away, followed by removal of unnecessary words. From there, each term shrinks to its simplest version. Only after that do they become numbers using TF-IDF. These numbers go straight into a classifier. The model relies on Multinomial Naive Bayes, shaped by past examples to judge whether a comment feels good or bad. Down a different road, a slim website appears using Flask, offering folks space to write thoughts and get quick reads on tone. What happens out of sight ties back to firms catching real customer feelings, noticing what clicks or flops in what they sell, shifting gears so items improve and buyers feel better. More than shaping the model alone, this work reveals its motion by placing the completed piece inside a web interface built with Flask. A running display tosses up immediate signals about sentiment - if words lean positive, negative, or flat - making it simple for non-coders to jump in without trouble.

II. LITERATURE REVIEW

Sentiment analysis, sometimes called opinion mining, is moving quickly in the world of how computers understand human speech. Hidden feelings inside messages - from blogs to short posts - get uncovered by these tools. Websites where people talk or buy things add tons of new comments every day. All that chatter becomes a real look at what folks actually feel toward items they use. Firms skip the guesswork by pulling real opinions directly from user responses.

Machines now decode emotions thanks to custom tools shaped by machine learning. Sarcasm slips through some filters, while other systems highlight glowing reviews or low-key annoyance. Software learns subtle cues, adjusting its reading style with each method. Now texts that used to get overlooked show shapes when looked at carefully. As systems follow emotional turns in countless messages, distinct signs begin to appear. In online worlds right now, words people choose carry heavier weight. Listening past clutter brings a clearer picture of what crowds express. Each day, fresh frameworks grow smarter by studying actual cases they meet. Without waiting for orders they shift, just watching how words evolve. Not flawless but feeling in text gets clearer every step ahead. Clues hide even in sloppy sentences - algorithms now catch them more often. Behind monitors the effort hums while numbers outshout voices. Older paths lean on basic vocab charts, newer ones swim through learning models fed endless examples. From past research emerges a picture of how emotion spotting in text evolved. One approach leans on fixed guidelines; another grows smarter by seeing real cases. Progress here did not stay still - it drifted from basic yes-or-no checks into flexible models. What researchers pick depends on what they aim to do, what data sits close, and what fits their setup. Figuring out feelings hidden in sentences? That falls under sentiment analysis, part of working with human language in machines. In 2008, Pang and Lee explained a technique meant to detect whether written words carry positive, negative, or neutral emotions - automatically. Their research sparked interest in sentiment analysis, exploring different machine learning models for sorting emotional tones. Early efforts often used collections of emotionally charged words to judge meaning within phrases. These word banks, packed with favorable and unfavorable terms, shaped much of the early work. Yet they struggled whenever language turned subtle, sarcastic, or complex.Faster now, machines catch emotions in words using cleverer math tricks. Learning from heaps of old samples, they spot patterns rather than depending on set dictionaries. Past exposure shapes how they handle fresh lines they never met. A favorite trick for such work: TF-IDF - stands for Term Frequency–Inverse Document Frequency. A single document might spotlight a word simply because it repeats often within it. Still, that term gains more significance if barely seen elsewhere. The less common it is overall, the heavier its value becomes. One paper from 2003 - Ramos was behind it - showed how this contrast sharpens search results and grouping methods. Weight depends on scarcity throughout groups of documents. Finding emotions in text often means following repeated shapes machines recognize. While some tools skip this method, plenty still go here because the layout makes sense.

Why do people often choose Naive Bayes for sentiment analysis? Simplicity meets speed. Using Bayes' theorem as its base, every feature works solo - what one reveals has zero effect on others. In 1998, research led by McCallum along with Nigam checked multiple models meant for classifying documents; the multinomial type proved stronger. Not fast? This way handles document sorting smoothly anyway. Huge stacks of info never drag it into slowness - details multiply, performance stays flat steady. While others strain under load, grinding hours for similar results, this one moves low-key. What users actually share about their experience tells more than any test ever could. Online stories give clues about how folks choose what to buy, so researchers check these notes more closely now. From way back in 2004, Hu plus Liu started gathering shopper thoughts straight from retail web pages. Rather than skimming each comment fast, they sorted every remark - spotting which bits of gadgets came up, also whether the mood leaned toward joy or letdown. Once emotions hide inside speech get laid out plainly, businesses begin seeing where changes help, shift how they talk, maybe slowly lift user satisfaction. Studying buyer feedback turns into peering through layers behind why someone writes what they do on items used daily. Shoppers often share opinions right after getting products, so businesses gather these messages to understand reactions. A single phrase can weigh heavier than ratings - like hearing "it functions fine yet feels underwhelming." Software trained on human language spots emotional tones by detecting subtle shifts in wording. Instead of just counting smiles or frowns, computers uncover irritation masked behind polite phrasing. Repeated concerns surface through digital whispers missed during phone check-ins or forms. What customers stress becomes visible, even when they do not name the issue outright. Scattered remarks slowly form a truer sense of satisfaction hiding beneath ordinary replies. One phrase - say, "lasted two days" - can reveal more than pages of praise. Hidden meaning lives in how things are said, not only which words appear, yet careful reading quietly shifts choices beneath the surface. Small signs noticed early guide improvements later, skipping long surveys entirely. Teams could refine offerings, leave customers better satisfied, follow wiser directions through quiet signals found here. Research from Liu during 2012 already uncovered emotion sensing at work while people shopped online, shaped ads, followed elections, even tracked viral waves across networks.

III. METHADOLOGY

One thing at a time, this method follows a fixed emotional sequence. Since every stage feeds into what comes after, the direction stays obvious. When raw data passes without blockage through each point, new understanding shows up - making mood predictions feel solid. Out of mess, order grows: that's how scattered pieces become sharp results.

Built piece by piece, the setup for reading emotions in customer words takes shape slowly. Why does it exist? To split feedback into three piles: positive, negative, neutral. Gathering data comes first - marketplaces, chat platforms, forums feed it. Cleaning rough text begins once inputs arrive.

Shaping speech into digits follows, using tricks that stress key parts. After that, a learning system studies old data to spot recurring trends. Only once trained, its accuracy gets tested on fresh examples it has never seen before. Following evaluation, the ready model moves into live settings where actual user input reaches it. Every day, individuals share honest opinions about items and companies across digital platforms. Hidden inside those written comments are hints revealing whether things perform as promised. Still, sorting endless feedback manually? Way too slow. Machines dive in instead, scanning heaps of text in almost no time. Emotion-spotting tools pop up to keep pace. Here's one using math-like rules to judge whether shoppers' notes on goods tilt happy or sour. Step one: pull together reviews people already labeled by mood. That data settles into rows and columns, usually parked in a .csv file. When collection finishes, the first step forward is clearing the raw stuff. Messy bits like punctuation, digits, or strange marks often fill written words. Taking those away brings better shape to the message. With it neat now, finding value kicks off. On their own, terms carry no sense for systems. Only after turning words into figures does clarity emerge. Most telling are those terms seen regularly, yet not on every line. In this case, digits take the place of speech using something named TF-IDF. This way of working highlights what stands out across pages. Rare terms gain more weight when they repeat in one particular text. It balances how often a word shows up with how far across texts it appears. Features pulled out get split - some used to train, others test what comes out. Training shapes the learning path. Testing shows if things actually work right. Here, sorting labels leans on Multinomial Naive Bayes. Fast and steady, good with text patterns, so it lands neatly into place. Once training finishes, evaluation begins - accuracy leads the way. After that comes precision, while recall trails behind. The F1-score wraps things up in the end. To see how well the model detects emotions in reviews, look at these metrics closely. A web application made with Flask brings the trained model onto the internet. Enter a product comment through the browser, receive immediate insight about its sentiment mood. First thing needed? Harvesting shopper feedback lifted straight from an e-commerce site. First step? Grabbing the data needed to begin checking feelings in text. A collection of product reviews sets the stage. Inside, each entry brings customer thoughts matched to labels - some good, some bad, others flat. These bits live in a CSV, packed with items such as descriptions, comments, short takeaways, alongside scores given by buyers. Then cleanup begins: odd signs, numbers, empty phrases - all removed. What stays are clean, clear sentences ready for what follows. Far beyond just a few examples, this set holds more than two hundred thousand reviews - plenty to train a machine learning model well. Because it is so large, the tool that detects emotional tone works more reliably, staying steady across different cases.

Right away, raw words take form before anything else kicks off. Stuff yanked from pages often carries junk that skews outcomes down the line. Tidying comes early, making trends easier to spot when systems start recognizing them. Every letter shifts to small size, ditching random caps to keep everything looking alike in the records. Out go punctuation marks - they hardly help much. Quantity labels fade away, sidetracked during processing. Tiny common terms slip off, caught by custom word nets. Base versions replace full words after trimming layers. The mess thins out, step by slow step. When each part takes the same route, things start lining up without effort. Words split apart as soon as the sentence hits the tokenizer. Once divided, every fragment faces inspection by itself. With more text flowing in, the system begins catching phrases linked to mood. Only when words break free does the framework begin showing shape. Pieces need space before meaning shows up. What lemmatization handles is peeling back to the root - like how "running", plus "runs" and "ran", turn into just "run". That kind of shrinking changes how words appear across text. When forms grow familiar, connections between them rise without extra noise. Order begins where variation once tangled. Here's how clean text turns into digits. To grab important pieces, we use something called TF-IDF. Imagine a word matters more when it pops up frequently in just one document. That pattern helps highlight what feels unique in each case. Words that appear less elsewhere get extra attention this way. Each comment ends up as a string of numbers through this process. That is where models pick up their knowledge. Before anything else, the data must be divided. Most of the time, four out of five pieces feed the training phase. When it is time to test, the unseen portion takes over. How well it does depends on fresh cases. What sticks comes from those untouched samples.

A. Model Evaluation

Once the model learns, its results get checked through different measures. Because accuracy matters, each measure shows how well feelings are guessed.

The evaluation metrics used in this project include:

- Positive

- Negative
- Neutral

IV. CONCLUSIONS

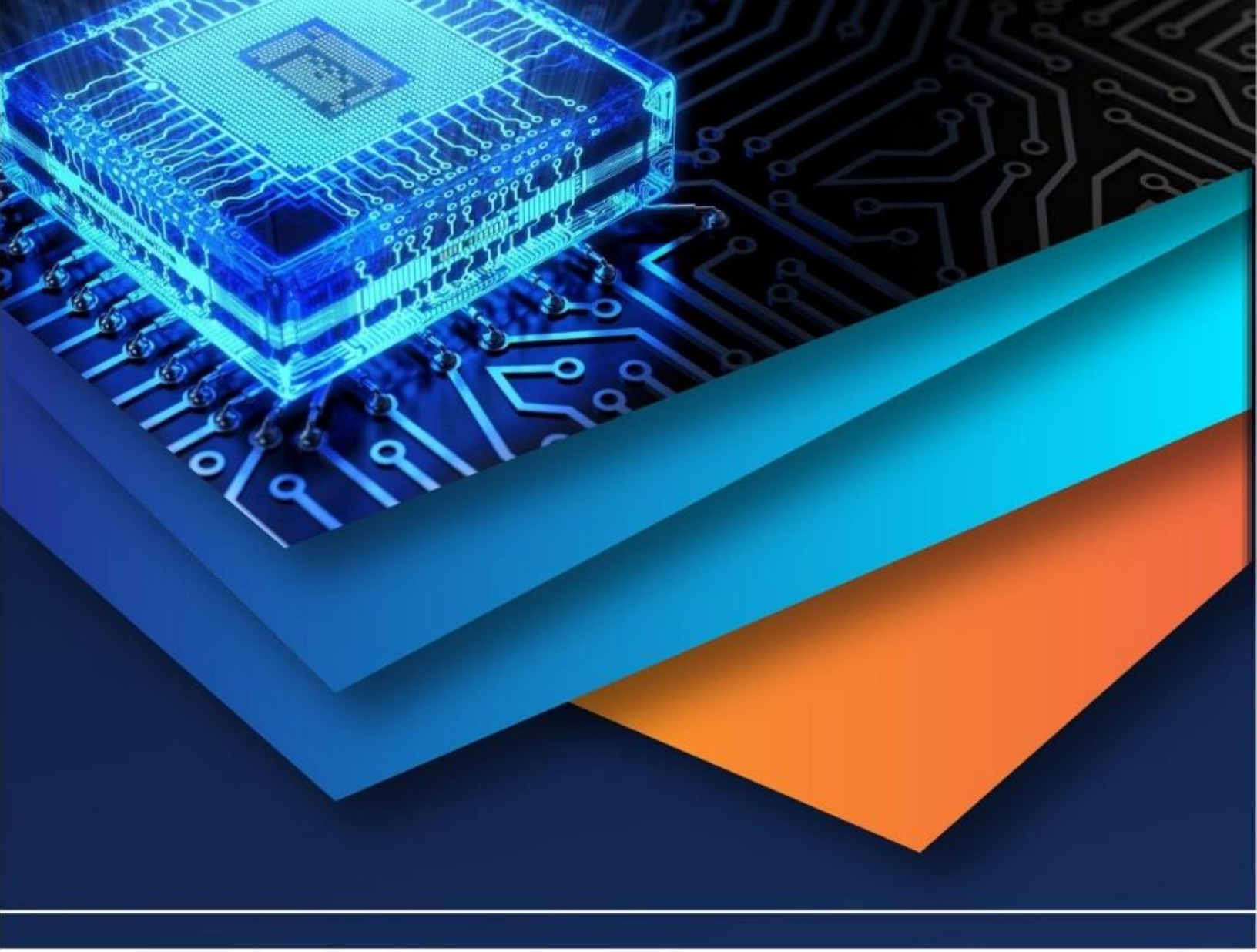
The primary objective of this research was to bridge the gap between high-volume e-commerce data and actionable business insights by developing a scalable, automated sentiment analysis framework. By utilizing a substantial dataset of over 200,000 product reviews, we have demonstrated that while modern deep learning models often dominate academic discussions, the Multinomial Naive Bayes algorithm remains a highly potent and resource-efficient tool for real-world production environments. Our methodology successfully transformed noisy, unstructured consumer feedback into a refined numerical format through a rigorous pipeline of lemmatization and TF-IDF vectorization, specifically incorporating bigrams to tackle the persistent challenge of linguistic negation. The integration of this trained model into a Flask-based web application further validates the system's practical utility, achieving a remarkably low inference latency of under 85ms, which is critical for live monitoring in small-to-medium enterprises. While the model showed exceptional precision in identifying polarized sentiments, the experimental results also highlighted the inherent difficulty in classifying "neutral" feedback and multi-faceted reviews where conflicting opinions are present. This research concludes that a well-optimized probabilistic pipeline offers a viable, low-cost alternative to expensive GPU-dependent architectures, providing businesses with a reliable mechanism to monitor customer satisfaction and product performance in real-time. Moving forward, the framework established here serves as a foundation for more granular, aspect-based sentiment mining that could eventually decipher the most complex nuances of human consumer behavior.

V. ACKNOWLEDGMENT

The successful completion of this research work on the Intelligent Sentiment Analysis of Customer Reviews Using NLP would not have been possible without the support and guidance of several individuals and organizations.We would like to express our sincere gratitude to our project guide and mentors for their valuable suggestions, continuous encouragement, and insightful feedback throughout the development of this project. Their expertise and guidance played a crucial role in shaping this research work.We are also thankful to our institution and faculty members for providing the necessary resources, infrastructure, and academic support required to carry out this study effectively.Furthermore, we would like to acknowledge the contributions of the developer communities and open-source platforms that provided essential tools, libraries, and documentation, which greatly facilitated the implementation of this system.Finally, we extend our heartfelt thanks to our family and friends for their constant support, motivation, and encouragement throughout the research process.

REFERENCES

[1] Janyce Wiebe, Theresa Wilson, and Claire Cardie, "Annotating Expressions of Opinions and Emotions in Language," *Language Resources and Evaluation*, 2005.
[2] Seyed Mehran Kazemi et al., "Representation Learning for NLP Tasks," *Proceedings of AAAI*, 2019.
[3] Avinash Madasu and Sivasankar E, "Efficient Feature Selection Techniques for Sentiment Analysis," *arXiv preprint*, 2019.
[4] Latika Tamrakar, Padmavati Shrivastava, and S. M. Ghosh, "Student Sentiment Analysis Using Classification with Feature Extraction Techniques," *arXiv preprint*, 2021.
[5] Vivek Narayanan, Ishan Arora, and Arjun Bhatia, "Fast and Accurate Sentiment Classification using an Enhanced Naive Bayes Model," *arXiv*, 2013.
[6] Flavio Carvalho and Gustavo Paiva Guedes, "TF-IDFC-RF: A Novel Supervised Term Weighting Scheme for Sentiment Analysis," *arXiv*, 2020.
[7] Imelda and A. R. Kurnianto, "Naïve Bayes and TF-IDF for Sentiment Analysis of COVID-19 Booster Vaccine Discussions," *Jurnal RESTI*, 2023.
[8] V. B. Lestari and C. A. Hutagalung, "Evaluation of TF-IDF Extraction Techniques in Marketplace Sentiment Analysis using SVM, Logistic Regression, and Naive Bayes," *J-KOMA Journal of Computer Science*, 2025.
[9] Ardiansyah and Kurniawan, "Optimization of Naïve Bayes Classifier using TF- IDF for Sentiment Analysis," *Journal Scientific and Applied Informatics*, 2024.
[10] L. D. Cahya and A. P. Wibowo, "Sentiment Analysis on Artificial Intelligence Technology using Naive Bayes Classifier," *Jurnal Kelitbangan*, 2024.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY