

Context-Aware Sentiment Intelligence and Personalized Product Suggestions Using Machine Learning

Mrs. B. Lakshmidevi HOD/CSE(SC)¹, Gopiga S², Monica M³, Lavanya S⁴

^{1,2,3,4} *Department of Computer Science and Engineering, Surya Group of Institutions, Anna University*
doi.org/10.64643/IJIRTV12111-200951-459

Abstract—This project presents a context-aware sentiment analysis and brand popularity monitoring system for digital commerce platforms. The system collects customer reviews and social media opinions from platforms like Amazon, Flipkart, Twitter, and Instagram. Using Natural Language Processing (NLP) techniques and a fine-tuned BERT model, it classifies sentiments accurately as positive, negative, or neutral. The system performs aspect-based sentiment analysis to identify product-level strengths and weaknesses. It extracts meaningful insights from large volumes of unstructured text data. These insights help businesses understand customer expectations and market trends. An interactive dashboard built using Flask and MySQL visualizes brand popularity scores, sentiment trends, and engagement metrics in real time. The system supports manufacturers, retailers, and customers with intelligent product recommendations. Overall, the project provides an automated, scalable, and intelligent decision-support solution for e-commerce analytics.

Index Terms—Sentiment Analysis, BERT, NLP, Brand Popularity Analytics, Aspect-Based Analysis, Digital Commerce, Machine Learning, Recommendation System, Data Mining, Dashboard Analytics.

I. INTRODUCTION

The rapid growth of digital commerce has resulted in massive volumes of customer reviews, ratings, and social media feedback across platforms such as Amazon, Flipkart, Twitter, and Instagram. Every day, thousands of users share their opinions about products and brands. For businesses, manually analyzing this huge amount of unstructured textual data is time-consuming, inefficient, and impractical. Therefore, there is a strong need for automated systems that can process, analyze, and interpret customer feedback accurately and in real time.

Sentiment analysis, a major application of Natural Language Processing (NLP) and Machine Learning, is widely used to extract opinions and emotional tone from textual data. Traditional techniques such as lexicon-based approaches and classical machine learning algorithms (Naïve Bayes, SVM, Logistic Regression) often fail to understand contextual meaning, sarcasm, slang, and complex sentence structures. To overcome these limitations, transformer-based deep learning models such as BERT (Bidirectional Encoder Representations from Transformers) are used. BERT enables deep contextual understanding by analyzing words based on surrounding context, significantly improving sentiment classification accuracy.

This project proposes a context-aware consumer opinion and brand popularity analytics system that integrates multi-platform review data into a unified framework. The system uses NLP preprocessing techniques such as tokenization, stop-word removal, text normalization, and cleaning to prepare the data. A fine-tuned BERT model is applied for sentiment classification, while aspect-based sentiment analysis identifies product-specific strengths and weaknesses. The system also calculates brand popularity metrics using engagement levels, review volume, and sentiment scores.

For implementation, technologies such as Python, Flask framework, MySQL database, and Bootstrap are used to build a responsive web-based dashboard. Libraries like Pandas, NumPy, Scikit-learn, Transformers, and TensorFlow/PyTorch support data processing and model training. The final output is displayed through interactive dashboards showing sentiment trends, brand rankings, and personalized product recommendations. This system helps manufacturers, retailers, and customers make data-driven decisions, improve product quality, and enhance digital marketing strategies.

A. Problem Statement

In today's digital commerce environment, businesses receive thousands of customer reviews, comments, ratings, and social media mentions every day across multiple platforms such as e-commerce websites and social networking sites. These reviews contain valuable insights about product quality, customer satisfaction, and brand reputation. However, analyzing such a massive volume of unstructured textual data manually is highly inefficient and time-consuming. As a result, many businesses fail to fully utilize customer feedback for strategic decision-making and product improvement.

Existing sentiment analysis systems primarily rely on traditional machine learning algorithms or lexicon-based methods. These approaches treat words independently and lack deep contextual understanding. They struggle to interpret slang, sarcasm, mixed sentiments, emojis, and complex sentence structures commonly used in online reviews. Furthermore, many systems operate on a single platform and do not provide a unified view of brand perception across multiple digital channels. This leads to incomplete or inaccurate analysis results.

There is therefore a strong need for a unified, intelligent, and automated system capable of collecting multi-platform review data and analyzing it with high contextual accuracy. This project proposes the use of advanced Natural Language Processing (NLP) techniques along with transformer-based models like BERT to understand the contextual meaning of customer opinions. NLP preprocessing methods such as tokenization, normalization, stop-word removal, and noise filtering can be used to prepare raw textual data for effective analysis. Aspect-based sentiment analysis can further identify product-level strengths and weaknesses.

To implement this solution, technologies such as Python for programming, Flask for backend development, MySQL for database management, and Bootstrap for frontend design can be used. Machine learning libraries like Pandas, NumPy, Scikit-learn, and Transformers (with TensorFlow or PyTorch) can support model training and deployment. The system can also include an interactive dashboard to display sentiment trends, brand popularity scores, and personalized recommendations. This comprehensive approach ensures accurate sentiment detection, real-

time insights, and improved decision-making for businesses and customers.

B. Objectives

To Develop a BERT-Based Sentiment Analysis System for Multi-Platform E-Commerce Data :The primary objective of this project is to design and implement a sentiment analysis system using a fine-tuned BERT (Bidirectional Encoder Representations from Transformers) model. BERT is chosen because of its deep contextual understanding capability, which helps interpret complex customer reviews accurately. The system collects review data from multiple platforms such as Amazon, Flipkart, Twitter, and Instagram, creating a unified analysis framework. Python programming language, along with the Transformers library (using TensorFlow or PyTorch), can be used to train and deploy the BERT model effectively.

To Preprocess and Clean Textual Data:Before performing sentiment analysis, raw textual data must be cleaned and standardized. This objective focuses on applying Natural Language Processing (NLP) preprocessing techniques such as tokenization, stop-word removal, lowercasing, punctuation removal, stemming or lemmatization, and noise filtering. Libraries such as NLTK, SpaCy, and Pandas can be used for efficient text preprocessing. Proper cleaning improves model accuracy by removing irrelevant or misleading information from customer reviews.

To Classify Sentiments Using Fine-Tuned BERT:This objective involves training and fine-tuning the BERT model on labeled sentiment datasets to classify reviews into positive, negative, or neutral categories. The BERT tokenizer converts text into embeddings that capture contextual meaning. The model can be fine-tuned using supervised learning techniques and evaluated using performance metrics such as accuracy, precision, recall, and F1-score. Deep learning frameworks like TensorFlow or PyTorch can be used for model training and optimization.

To Perform Aspect-Based Sentiment Analysis:In addition to overall sentiment classification, the system aims to identify specific product features (aspects) mentioned in reviews, such as price, quality, delivery, or customer service. Aspect extraction techniques using NLP and dependency parsing can be implemented to detect feature-level opinions. This

helps businesses understand which specific areas need improvement. Libraries like SpaCy or custom deep learning models can be used for aspect detection and sentiment scoring.

To Monitor Brand Popularity in Real Time:The system calculates brand popularity by analyzing review volume, engagement metrics, sentiment distribution, and keyword frequency. Real-time data processing can be achieved using backend technologies like Flask and MySQL for storing and retrieving sentiment results. Visualization tools such as Matplotlib or interactive dashboards built with Bootstrap can display brand rankings and trend graphs. This enables businesses to track reputation changes instantly.

To Provide Personalized Product Recommendations:Based on sentiment outcomes and popularity scores, the system generates personalized recommendations for different users such as manufacturers, retailers, and customers. Recommendation logic can be implemented using rule-based filtering or machine learning techniques. Python libraries like Scikit-learn can support recommendation modeling. This module enhances decision-making by suggesting product improvements, stock strategies, or purchase options according to real-time customer feedback.

C. Organization of Paper

This paper is systematically organized to clearly explain the development of the Context-Aware Consumer Opinion and Brand Popularity Analytics System. The first section introduces the background of digital commerce growth, the importance of sentiment analysis, and the motivation behind the project. It also presents the problem statement and objectives of building a BERT-based sentiment analysis system. This section explains the use of Natural Language Processing (NLP), Machine Learning, and transformer-based models like BERT to handle large-scale customer review data effectively.

The next section discusses the related work, which reviews existing research papers and previously developed systems in sentiment analysis, brand analytics, and recommendation systems. It compares traditional methods such as lexicon-based approaches and classical machine learning models with advanced deep learning techniques. This section highlights how

modern transformer models, multimodal learning, and big data analytics improve accuracy and contextual understanding. It helps justify the need for implementing BERT and aspect-based sentiment analysis in this project.

The system analysis and system design sections provide a detailed explanation of the architecture and workflow of the proposed system. These sections describe the existing system limitations, proposed improvements, and overall system architecture. Technologies such as Python, Flask framework, MySQL database, and Bootstrap are explained in terms of their roles in backend development, database management, and frontend interface design. The modules such as NLP preprocessing, BERT sentiment classification, brand popularity monitoring, and recommendation generation are also described in detail.

The implementation and result analysis sections explain how the system is developed and tested using tools and libraries like Pandas, NumPy, Scikit-learn, Transformers, and TensorFlow/PyTorch. It discusses hardware and software requirements, model training, performance evaluation, and dashboard visualization. Finally, the conclusion section summarizes the project achievements, including high-accuracy sentiment classification and real-time brand analytics, and suggests future enhancements such as multilingual sentiment analysis and multimodal data integration.

II. RELATED WORK

Several research studies have focused on sentiment analysis, opinion mining, and brand analytics in digital commerce and social media environments. Early approaches primarily used lexicon-based methods, where predefined sentiment dictionaries were applied to classify text as positive or negative. Later, traditional machine learning algorithms such as Naïve Bayes, Support Vector Machines (SVM), and Logistic Regression were introduced to improve classification accuracy. These models relied heavily on manual feature engineering techniques such as TF-IDF and Bag-of-Words, which often failed to capture contextual meaning in complex sentences.

With the advancement of deep learning, researchers began adopting neural network-based models such as Recurrent Neural Networks (RNN), Long Short-

Term Memory (LSTM), and Convolutional Neural Networks (CNN) for sentiment analysis. These models improved performance by capturing sequential dependencies in text data. More recently, transformer-based architectures such as BERT (Bidirectional Encoder Representations from Transformers) have significantly enhanced contextual understanding by analyzing words based on both left and right context. BERT and its variants achieve higher accuracy and better handling of sarcasm, slang, and domain-specific language.

In addition to text-based sentiment analysis, recent research has explored multimodal learning approaches that combine textual data with images, logos, and visual content. Vision-Language models such as CLIP and ViT are used to detect brand-related patterns and analyze product representation across platforms. Big data analytics and social media mining techniques are also used to track brand reputation trends and consumer engagement metrics. However, many of these studies focus only on model performance and do not integrate user-friendly dashboards or real-time monitoring systems.

Despite the advancements in sentiment classification accuracy, several existing systems lack unified multi-platform integration, real-time analytics, and personalized recommendation modules. This project addresses these research gaps by integrating NLP preprocessing, fine-tuned BERT-based sentiment classification, aspect-based sentiment analysis, and brand popularity monitoring within a single framework. Technologies such as Python, Flask, MySQL, Transformers, and visualization tools like Matplotlib or Bootstrap dashboards can be used to build an interactive and practical system that not only analyzes sentiment but also provides actionable insights and product recommendations for different user roles.

III. SYSTEM ANALYSIS

A. Existing System

Existing sentiment analysis systems primarily rely on lexicon-based approaches and traditional machine learning algorithms such as Naïve Bayes, Support Vector Machines (SVM), and Logistic Regression. Lexicon-based methods use predefined sentiment

dictionaries where words are assigned positive or negative scores. Classical ML models use feature extraction techniques like Bag-of-Words (BoW) and TF-IDF to convert text into numerical format. These approaches are relatively simple to implement using tools such as Scikit-learn and basic NLP libraries, but they often provide limited accuracy in real-world applications.

One of the major limitations of these systems is that they treat words independently without understanding the overall context of a sentence. For example, they may misinterpret sarcasm, negations, or domain-specific slang commonly used in online reviews. Since these models rely heavily on manual feature engineering, their performance depends on carefully designed input features. This makes them less adaptable to different domains or evolving customer language patterns across platforms like Amazon or social media.

Early neural network models such as Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) networks were introduced to improve contextual understanding. These models capture sequential relationships between words better than traditional algorithms. They can be implemented using deep learning frameworks such as TensorFlow or PyTorch. However, even these models struggle with long-term dependencies in lengthy reviews and require large computational resources and training data to achieve good performance.

Furthermore, most existing systems are platform-specific, meaning they analyze reviews from a single source and do not integrate data from multiple digital platforms. They often lack real-time dashboard visualization and unified brand comparison features. There is limited support for aspect-based sentiment analysis or recommendation systems. As a result, businesses do not receive comprehensive insights into brand popularity trends or actionable suggestions for product improvement and marketing strategies.

B. Drawbacks

Poor Contextual Understanding: Traditional sentiment analysis systems rely on lexicon-based methods or basic machine learning models that treat words independently without understanding their

surrounding context. These models cannot interpret the real meaning of sentences where word sentiment changes based on context. For example, phrases with negation like “not good” may be incorrectly classified as positive. To overcome this issue, advanced transformer-based models such as BERT can be used, as they analyze text bidirectionally and understand contextual relationships between words more effectively.

Cannot Detect Sarcasm or Slang: Online reviews often contain sarcasm, informal language, emojis, and slang expressions. Classical ML algorithms like Naïve Bayes and SVM fail to interpret such expressions correctly because they depend on simple word frequency features. For instance, a sarcastic sentence like “Great job ruining the product!” may be misclassified as positive. Deep learning models such as BERT, combined with proper NLP preprocessing techniques (tokenization, normalization, emoji handling), can better capture tone and contextual nuance in customer feedback.

Requires Manual Feature Engineering: Traditional machine learning approaches require manual extraction of features such as TF-IDF or Bag-of-Words. This process is time-consuming and domain-dependent, meaning the system must be redesigned for different types of products or industries. Manual feature engineering also limits scalability. In contrast, transformer-based architectures automatically generate contextual embeddings, reducing human intervention and improving adaptability across various domains of digital commerce.

Low Accuracy for Complex Reviews: Customer reviews are often lengthy and contain mixed sentiments about multiple product aspects. Classical models struggle with long and complex sentences, leading to reduced accuracy. Even early neural networks like RNN or LSTM have limitations in capturing long-term dependencies. Fine-tuned BERT models, implemented using TensorFlow or PyTorch, can handle long text sequences more efficiently and provide higher classification accuracy for real-world e-commerce reviews.

No Real-Time Brand Popularity Monitoring: Many existing systems focus only on sentiment

classification and do not provide real-time monitoring of brand popularity across multiple platforms. They lack visualization tools and unified dashboards. Without proper integration using backend technologies like Flask and databases such as MySQL, businesses cannot track sentiment trends dynamically. Implementing interactive dashboards with Bootstrap and visualization libraries like Matplotlib can enable real-time brand comparison and trend analysis.

No Personalized Recommendation Module: Traditional sentiment analysis systems only classify reviews but do not generate actionable recommendations. There is no mechanism to suggest product improvements for manufacturers or purchasing guidance for customers. A recommendation module can be developed using rule-based logic or machine learning algorithms with Scikit-learn. By integrating sentiment outcomes with brand popularity metrics, the system can provide personalized suggestions, making it more practical and business-oriented.

C. Proposed System

The proposed system is a unified and intelligent platform that integrates review and feedback data from multiple e-commerce and social media platforms such as Amazon, Flipkart, Twitter, and Instagram. Instead of analyzing a single data source, the system consolidates multi-platform information to provide a comprehensive view of customer opinions and brand perception. Data collection can be performed using datasets, APIs, or web scraping techniques, and the collected data is stored in a structured format using a MySQL database for efficient retrieval and processing.

To ensure accurate sentiment analysis, the system applies Natural Language Processing (NLP) preprocessing techniques such as tokenization, stop-word removal, normalization, lowercasing, and noise filtering. Libraries like NLTK, SpaCy, and Pandas can be used for text cleaning and preparation. After preprocessing, a fine-tuned BERT model (implemented using the Transformers library with TensorFlow or PyTorch) is used for sentiment classification. BERT captures contextual meaning by analyzing words bidirectionally, enabling accurate

classification of positive, negative, or neutral sentiments even in complex reviews.

In addition to overall sentiment detection, the system performs aspect-based sentiment analysis to identify specific product features mentioned in reviews, such as price, quality, delivery, packaging, or customer service. This helps businesses understand which particular aspects influence customer satisfaction. The brand popularity analyzer further calculates metrics such as review count, engagement level, keyword frequency, and sentiment distribution to generate a real-time brand popularity index.

The final output is presented through an interactive web-based dashboard developed using Flask for backend processing and Bootstrap for frontend design. Visualization tools such as Matplotlib or other charting libraries display sentiment trends, brand comparisons, and performance graphs. A recommendation module, developed using rule-based logic or machine learning techniques with Scikit-learn, provides intelligent suggestions for manufacturers, retailers, and customers. This integrated system ensures accurate analysis, real-time monitoring, and data-driven decision-making in digital commerce.

D. Advantages

High Accuracy Using BERT: One of the major advantages of the proposed system is its high sentiment classification accuracy achieved through the use of a fine-tuned BERT (Bidirectional Encoder Representations from Transformers) model. Unlike traditional machine learning algorithms, BERT understands contextual relationships between words by processing text bidirectionally. It can be implemented using the Hugging Face Transformers library with TensorFlow or PyTorch. This improves prediction accuracy for positive, negative, and neutral sentiments, especially in complex and real-world e-commerce reviews.

Context-Aware Sentiment Detection: The system performs context-aware sentiment analysis, meaning it understands the meaning of a sentence based on surrounding words rather than analyzing each word independently. This helps in correctly interpreting negations, sarcasm, slang, and domain-specific expressions commonly found in online reviews. NLP

preprocessing techniques such as tokenization, normalization, and stop-word removal (using NLTK or SpaCy) further enhance contextual understanding. As a result, the system provides more reliable and realistic sentiment predictions.

Multi-Platform Data Integration: Another important advantage is the integration of data from multiple platforms such as e-commerce websites and social media. Instead of analyzing a single data source, the system consolidates review data into a unified database using MySQL. This enables comprehensive brand comparison and consistent sentiment tracking across platforms. Python-based data processing tools like Pandas and NumPy can be used to manage and analyze large datasets efficiently.

Real-Time Dashboard Visualization: The system includes an interactive web-based dashboard that visually represents sentiment trends, brand rankings, and engagement metrics. The dashboard can be developed using Flask for backend processing and Bootstrap, HTML, and CSS for frontend design. Visualization libraries such as Matplotlib can generate charts and graphs for better data interpretation. This real-time monitoring capability helps businesses make quick and informed decisions.

Automated Large-Scale Processing: The proposed system is designed to handle large volumes of customer reviews automatically. Once the model is trained, it can process thousands of reviews in real time without manual intervention. Automation is achieved using Python scripts, backend APIs, and database integration. This reduces human effort, saves time, and increases operational efficiency for businesses managing large-scale digital commerce platforms.

Personalized Recommendations: In addition to sentiment analysis, the system provides personalized recommendations for different user roles such as manufacturers, retailers, and customers. The recommendation module can be implemented using rule-based filtering or machine learning algorithms with Scikit-learn. By analyzing sentiment scores and brand popularity metrics, the system suggests product improvements, stocking strategies, or purchase options. This makes the project not only analytical but also practically beneficial for business growth.

IV. SYSTEM DESIGN

A. System Architecture

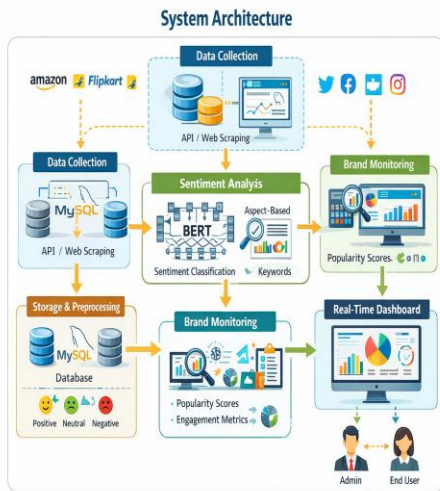


Figure 1 : System Architecture

B. Module Description

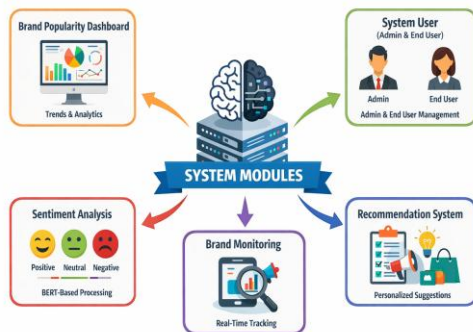


Figure 2 : Module Description

C. Data Flow Diagram

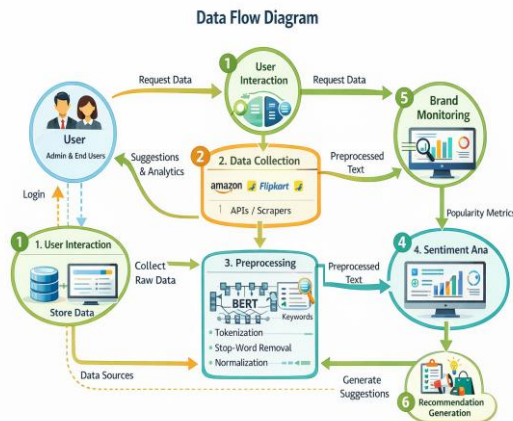


Figure 3: Data Flow Diagram

D. DataBase Diagram

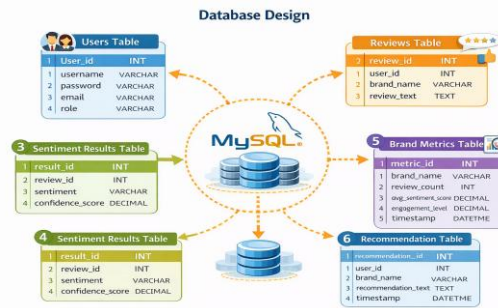


Figure 4 : Database Diagram

V. IMPLEMENTATION

A. Hardware Requirements

The proposed sentiment analysis and brand popularity monitoring system requires a minimum Intel i5 processor or higher to efficiently handle computational tasks. Since the project involves Natural Language Processing (NLP) and a transformer-based deep learning model like BERT, strong processing capability is essential. The processor supports model training, real-time sentiment prediction, data preprocessing, and backend operations developed using Python and Flask. A multi-core processor ensures faster execution and smooth multitasking during large-scale review analysis.

A minimum of 16 GB RAM is recommended to manage large datasets and deep learning workloads effectively. During preprocessing, thousands of customer reviews may be loaded into memory using libraries such as Pandas and NumPy. Additionally, frameworks like TensorFlow or PyTorch require significant memory resources when fine-tuning the BERT model. Adequate RAM prevents system lag, ensures stable model execution, and improves overall performance while handling high-volume multi-platform data.

The system also requires at least 512 GB SSD storage to store datasets, trained models, project source code, database files, and visualization outputs. Transformer-based models like BERT are large in size and require considerable disk space. Using an SSD instead of a traditional HDD significantly improves data read/write speed, which reduces loading time and enhances system responsiveness. Faster storage ensures efficient database operations

with MySQL and quick access to stored sentiment results.

Reliable internet connectivity is necessary for downloading pretrained BERT models from repositories such as Hugging Face, collecting review data via APIs or web scraping, and monitoring social media platforms in real time. Internet access also supports software updates, package installations, and cloud-based deployment if required. Stable connectivity ensures uninterrupted data integration and smooth functioning of the brand popularity analytics system.

B. Software Requirements

The primary programming language used in this project is Python 3.9, which provides strong support for machine learning, Natural Language Processing (NLP), and web development. Python is widely used in data science because of its simplicity, readability, and extensive library support. It is used for implementing text preprocessing, BERT-based sentiment classification, aspect extraction, and recommendation logic. Python also integrates smoothly with databases and web frameworks, making it ideal for building an end-to-end sentiment analytics system.

The Flask framework is used for backend development and creating the web application interface. Flask is a lightweight Python web framework that allows easy integration of machine learning models into a web-based system. It handles user authentication (Admin and End User), manages API requests, processes sentiment predictions, and connects the frontend with the database. Flask ensures smooth communication between the BERT model, MySQL database, and dashboard interface.

The project uses a MySQL database for storing user data, review data, sentiment results, brand metrics, and recommendation outputs. MySQL provides structured data management, fast querying, and secure storage. It maintains relationships between tables such as Users, Reviews, Sentiment Results, Brand Metrics, and Recommendations. Additionally, WAMP Server is used to create a local server environment for running the web application, managing the database, and testing the system during development.

Several important libraries are used to support data processing and machine learning tasks. Pandas and

NumPy are used for data manipulation and numerical operations. The Transformers library (from Hugging Face) is used to implement and fine-tune the BERT model. TensorFlow or PyTorch serves as the deep learning framework for model training and inference. Matplotlib is used to generate graphs and visualizations for sentiment trends and brand popularity analysis. Together, these tools enable the development of a powerful, efficient, and scalable sentiment analysis and recommendation system.

C. Implementation

The project is implemented using Python as the core programming language and the Flask framework for backend web development. Flask is used to create the web application interface, manage routing, handle user authentication (Admin and End User), and integrate the machine learning model with the frontend dashboard. The system follows a modular architecture where data collection, preprocessing, sentiment analysis, brand monitoring, and recommendation modules are interconnected. Review data collected from multiple platforms is stored in a MySQL database, which ensures structured storage and efficient retrieval of large volumes of textual information.

The first major step in implementation is data preprocessing using Natural Language Processing (NLP) techniques. Raw review text is cleaned by removing unwanted characters, punctuation, and special symbols. Tokenization is performed to split text into meaningful units, followed by stop-word removal and normalization to standardize the data. Libraries such as NLTK, SpaCy, Pandas, and NumPy are used for text processing and dataset handling. Proper preprocessing ensures that the input text is optimized for accurate sentiment classification.

The core of the system is the fine-tuned BERT model, implemented using the Transformers library with TensorFlow or PyTorch. The BERT tokenizer converts text into contextual embeddings, and the model classifies sentiments into positive, negative, or neutral categories. Performance metrics such as accuracy, precision, recall, and F1-score are used to evaluate the model. Additionally, aspect-based sentiment analysis is implemented to identify specific product features (e.g., quality, price, delivery) and assign sentiment scores to each aspect.

Finally, brand popularity metrics such as review count, average sentiment score, engagement level, and keyword frequency are calculated. These results are stored in the database and visualized through an interactive dashboard developed using Flask, HTML, CSS, Bootstrap, and Matplotlib. The dashboard displays sentiment trends, brand rankings, and personalized recommendations generated through rule-based or machine learning logic. This complete implementation ensures automated large-scale processing, real-time analytics, and practical decision-support for digital commerce platforms.

D. Algorithm

The first step of the algorithm involves collecting multi-platform review data from e-commerce websites and social media platforms. Data can be gathered using APIs, web scraping techniques, or pre-collected datasets. The collected reviews, ratings, and brand-related comments are stored in a structured format using a MySQL database. Python scripts are used to automate data collection and ensure proper storage. This step ensures that the system has sufficient and diverse input data for accurate sentiment and brand analysis.

In the second step, the collected text data undergoes preprocessing using Natural Language Processing (NLP) techniques. This includes removing special characters, punctuation, and unwanted symbols, converting text to lowercase, performing tokenization, removing stop words, and applying normalization or lemmatization. Libraries such as NLTK, SpaCy, Pandas, and NumPy are used to clean and prepare the text. After preprocessing, the cleaned text is passed through the BERT tokenizer, which converts sentences into contextual embeddings suitable for deep learning models.

The third and fourth steps involve performing sentiment classification using a fine-tuned BERT model implemented with the Transformers library using TensorFlow or PyTorch. The model predicts whether a review is positive, negative, or neutral. After sentiment prediction, aspect-based sentiment analysis is performed to identify specific product features (such as quality, price, or delivery) and assign sentiment scores to each aspect. This helps generate more detailed insights beyond overall sentiment classification.

In the final steps, the system calculates a brand popularity score based on metrics such as sentiment distribution, review count, engagement level, and keyword frequency. A recommendation module then generates personalized suggestions using rule-based logic or machine learning algorithms (Scikit-learn). The final results are displayed through an interactive dashboard built using Flask, HTML, CSS, Bootstrap, and Matplotlib. The dashboard visualizes sentiment trends, brand comparisons, and recommendation outputs, providing real-time decision support for businesses and customers.

VI. RESULT ANALYSIS

The experimental results show that the proposed system achieves higher sentiment classification accuracy compared to traditional machine learning models such as Naïve Bayes and SVM. This improvement is mainly due to the use of the fine-tuned BERT model, which understands contextual relationships between words rather than analyzing them independently. Performance metrics such as accuracy, precision, recall, and F1-score are used to evaluate the model. The results indicate that transformer-based models implemented using the Transformers library with TensorFlow or PyTorch significantly outperform classical approaches in handling complex and real-world e-commerce reviews.

Aspect-based sentiment analysis further enhances the depth of insight provided by the system. Instead of giving only an overall sentiment score, the system identifies specific product aspects such as quality, price, delivery, packaging, and customer service. By assigning sentiment values to each aspect, businesses can clearly understand product strengths and weaknesses. NLP libraries such as SpaCy and custom deep learning techniques are used to extract aspects and compute feature-level sentiment scores, resulting in more detailed and actionable analytics.

The brand popularity index is calculated using multiple parameters including sentiment distribution, review volume, engagement rate, and keyword frequency. These metrics are processed using Python data analysis tools like Pandas and NumPy and stored in the MySQL database. The real-time dashboard developed using Flask, Bootstrap, and Matplotlib visually presents brand rankings, sentiment trends,

and comparison graphs. This visualization improves usability and enables faster, data-driven decision-making for manufacturers, retailers, and customers.

The recommendation module adds practical value to the system by generating personalized suggestions based on sentiment analysis and brand popularity metrics. Rule-based logic or machine learning algorithms implemented using Scikit-learn are used to provide targeted recommendations. Experimental evaluation confirms that the integrated system not only improves classification accuracy but also enhances insight extraction, automation efficiency, and overall business strategy compared to traditional sentiment analysis systems.

VII. CONCLUSION

This project successfully implements a context-aware sentiment analysis and brand popularity monitoring system using a fine-tuned BERT model. By leveraging advanced Natural Language Processing (NLP) techniques and transformer-based deep learning, the system overcomes the limitations of traditional machine learning methods. The use of Python, the Transformers library, and TensorFlow or PyTorch enables accurate sentiment classification by understanding contextual meaning, sarcasm, and complex sentence structures commonly found in online reviews.

The system integrates multi-platform review data from e-commerce and social media sources into a unified framework. Text preprocessing techniques such as tokenization, stop-word removal, and normalization are applied using NLP libraries like NLTK and SpaCy. The cleaned data is processed through the BERT model to classify sentiments and perform aspect-based analysis, identifying detailed product-level insights such as quality, price, and service performance. This comprehensive analysis helps businesses gain a deeper understanding of customer opinions.

Real-time dashboards developed using Flask, MySQL, HTML, CSS, Bootstrap, and Matplotlib provide visual analytics such as sentiment trends, brand rankings, and comparative performance metrics. The brand popularity index, calculated using review volume, engagement level, and sentiment distribution, supports data-driven decision-making. The interactive interface ensures that administrators

and end users can easily interpret insights and monitor brand reputation effectively.

The recommendation module further enhances the practical value of the system by generating personalized suggestions for manufacturers, retailers, and customers. Using rule-based logic or machine learning techniques with Scikit-learn, the system offers actionable guidance for product improvement, stocking strategies, and purchase decisions. In the future, the system can be extended with multilingual sentiment analysis, multimodal data integration (text + images), and cloud-based deployment to improve scalability and global applicability.

REFERENCES

- [1] T. Zhou, "Improved Sales Forecasting Using Trend and Seasonality Decomposition with LightGBM," 2023 6th International Conference on Artificial Intelligence and Big Data (ICAIBD), IEEE, 2023.
- [2] G. Hu, P. Bhargava, S. Fuhrmann, S. Ellinger, and N. Spasojevic, "Analyzing Users' Sentiment Towards Popular Consumer Industries and Brands on Twitter," IEEE International Conference on Data Mining Workshops (ICDMW), pp. 381–388, 2017.
- [3] X. Jiang and Y. Xie, "Research on Social Media Marketing Strategy of the Milk Tea Industry," 2022 7th International Conference on Social Sciences and Economic Development (ICSSD 2022), 2022.
- [4] A. Makrides, D. Vrontis, and M. Christofi, "The Gold Rush of Digital Marketing: Assessing Prospects of Building Brand Awareness Overseas," Business Perspectives and Research, vol. 8, no. 1, pp. 4–20, Jan. 2020.
- [5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," Proceedings of NAACL-HLT, 2019.
- [6] T. Mikolov et al., "Efficient Estimation of Word Representations in Vector Space," Proceedings of ICLR Workshops, 2013.
- [7] A. Vaswani et al., "Attention Is All You Need," Advances in Neural Information Processing Systems (NeurIPS), 2017.

- [8] F. Pedregosa et al., “Scikit-learn: Machine Learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.