SentimentSphere: Sentiment Classification of E-commerce Reviews Using Textual Data Analysis

Saikat Das
Department of Computer Science and Engineering
Ahasanullah University Of Science and Technology
Student ID: 20210104158, Section: C2
Email: saikatdasmain47@gmail.com, Group No: 08

***Abstract*—This project implements a sentiment analysis system using Logistic Regression to classify comments as either positive or negative. The dataset consists of 3.6 million labeled comments, where sentiment values are represented as 1 (negative) and 2 (positive). The primary focus of this project is to preprocess raw text data, convert it into numerical representations, train a classification model, and evaluate its performance using standard evaluation metrics.**

Additionally, this project integrates web scraping techniques to automatically extract customer reviews from e-commerce websites. The scraped reviews undergo preprocessing to remove noise, handle missing data, and improve classification accuracy. A system is also implemented to detect and update errors dynamically, ensuring continuous improvement in sentiment classification. The final system is designed to provide real-time sentiment analysis based on user reviews, making it easier to analyze customer opinions and feedback automatically.

***Index Terms*—Sentiment Analysis, Logistic Regression, Text Classification, Machine Learning, Natural Language Processing, Data Preprocessing, Web Scraping, Error Handling.**

I. Introduction

Sentiment analysis, also known as opinion mining, is the process of identifying and extracting subjective information from textual data. It is widely used in industries such as customer feedback analysis, social media monitoring, brand management, and product review analysis. The goal of sentiment analysis is to classify text into predefined categories such as positive, negative, or neutral sentiments.

This project focuses on implementing a sentiment analysis model using Logistic Regression, a widely used classification algorithm. The dataset used in this study consists of 3.6 million labeled comments, where each comment is categorized as either negative (1) or positive (2). The project involves three key phases:

1) **Preprocessing:** Cleaning and preparing the text data by removing unnecessary characters, stop words, and normalizing words for better classification accuracy.
2) **Model Training:** Using Logistic Regression to classify text into positive or negative sentiments.
3) **Evaluation:** Assessing the performance of the model using standard classification metrics to ensure its effectiveness.

Furthermore, to enhance the practical usability of this sentiment analysis system, an automated web scraping module is developed to extract reviews from e-commerce websites. This ensures real-time sentiment analysis by dynamically fetching and processing new customer feedback. The system also includes an error detection and correction mechanism, which updates incorrect predictions and improves the model over time.

By combining machine learning, web scraping, and real-time data processing, this project aims to provide a comprehensive and efficient sentiment analysis system that can automatically analyze customer sentiments with high accuracy and minimal manual effort.

II. Literature Review

Sentiment analysis (SA) has gained significant attention in recent years, especially in e-commerce, where customer reviews contain valuable insights. Several studies have focused on different techniques to improve sentiment classification accuracy and efficiency.

Huang et al. [1] provided a comprehensive review of sentiment analysis techniques used in e-commerce platforms. Their study analyzed 271 research papers and found that 48% of them used machine learning, 44% applied deep learning, and 7% used a hybrid approach. The study also highlighted Amazon and Twitter as the most common sources of sentiment data. Future research directions include aspect-based sentiment analysis, sarcasm detection, and fine-grained sentiment classification.

Kaur et al. [2] conducted sentiment analysis on customer reviews collected from Flipkart. Their research categorized sentiments into positive, negative, and neutral classes. They also identified frequently used words in customer opinions, showing how sentiment words impact public perception. Their findings suggest that analyzing common negative and positive words can help understand customer psychology better.

Liu et al. [3] proposed a deep learning model, Bert-BiGRU-Softmax, for sentiment analysis of e-commerce product reviews. Their model integrates sentiment BERT for extracting features, a Bidirectional GRU for learning semantic relationships, and a Softmax layer for classification. The study, conducted on a dataset of over 500,000 reviews, achieved an accuracy of 95.5%, outperforming other models like RNN and BiLSTM.

Vijayaragavan et al. [4] introduced a novel Weighted Parallel Hybrid Deep Learning-based Sentiment Analysis on E-

Commerce Product Reviews (WPHDL-SAEPR). Their model combines a Restricted Boltzmann Machine (RBM) with Singular Value Decomposition (SVD) for improved sentiment classification. The approach enhances word representation using Word2Vec and refines sentiment classification through hybrid deep learning techniques. The study demonstrated high accuracy in classifying consumer sentiments from e-commerce reviews.

These studies highlight the growing importance of sentiment analysis in e-commerce. Machine learning and deep learning models continue to improve classification accuracy, making sentiment analysis more effective for businesses and consumers.

The reviewed literature highlights the significance of ML and DL models in sentiment analysis for e-commerce. Future research should focus on improving accuracy through hybrid models, better feature extraction, and context-aware sentiment classification.

III. Data Analysis

The dataset used in this project consists of Amazon customer reviews, with a total of 4 million entries, split into training and testing datasets.

A. Context

This dataset consists of 3.6 million Amazon customer reviews used for training the model, and 0.4 million Amazon customer reviews used for testing. Each review is labeled with a sentiment score, where the label "1" represents a negative sentiment, and "2" represents a positive sentiment.

B. Content

- **train.csv**: A CSV file containing 3.6 million customer reviews for training the model. Each review is labeled as either "1" (negative) or "2" (positive).
- **test.csv**: A CSV file containing 400,000 customer reviews for testing the model. The reviews are also classified into labels "1" (negative) or "2" (positive).

C. Preprocessing and Cleanliness of Data

Before performing the analysis, some preprocessing steps were applied to ensure the data is clean and ready for model training:

- Removal of duplicate entries and any irrelevant or incomplete data.
- Text normalization, including lowercasing all comments and removing punctuation.
- Tokenization of text, splitting the reviews into individual words or tokens for further analysis.

D. Insights and Patterns

Initial analysis indicates:

- The dataset is relatively balanced, with a roughly equal number of positive and negative reviews.
- Certain common words or phrases might frequently appear in positive or negative reviews, which can be valuable for feature extraction during model training.
- A deeper analysis can involve visualizing the distribution of sentiments and extracting frequent terms or n-grams that correlate with each sentiment class.

This preliminary data analysis helps understand the dataset's structure and cleanliness before moving to model training and evaluation. Further exploratory techniques can include visualizations and statistical tests to uncover more patterns.

IV. Methodology

A. Data Preprocessing

Data preprocessing is a critical step in any machine learning pipeline, especially when working with text data. The preprocessing steps involved in this study are as follows:

1) Data Cleaning: The raw text data often contains unwanted characters, punctuation, special symbols, and irrelevant information. To ensure high-quality input, the following steps were performed:

- **Removal of URLs**: Eliminated web links using regular expressions.
- **Removal of punctuation and special characters**: Stripped unnecessary symbols like !@#$%&̂*() to retain only meaningful words.
- **Removal of numbers**: Since numbers generally do not contribute to sentiment analysis, they were removed.
- **Lowercasing**: Standardized text to lowercase to prevent duplicate representations ("Text" and "text" are treated the same).

2) Tokenization: Each comment was split into individual words (tokens) to facilitate further processing.

3) Stopword Removal: Common words such as "is," "the," "and," "in" were removed as they do not carry significant meaning in sentiment analysis.

4) Stemming & Lemmatization: Words were reduced to their root forms to ensure uniformity:

- **Stemming**: Converted words like "running" to "run".
- **Lemmatization**: Mapped words like "better" to "good", preserving grammatical meaning.

B. Feature Engineering

After preprocessing, the cleaned text data was transformed into a numerical format suitable for training machine learning models.

1) TF-IDF Vectorization: The Term Frequency-Inverse Document Frequency (TF-IDF) method was used to convert text into numerical features.

- **Term Frequency (TF)**: Measures how often a word appears in a document.
- **Inverse Document Frequency (IDF)**: Reduces the importance of common words across multiple documents.
- **Why TF-IDF?** Unlike simple word frequency counts, TF-IDF highlights important words that appear frequently in a document but rarely in the entire dataset.
- **Implementation**: The text was transformed into a TF-IDF matrix with a maximum of 5000 features, capturing unigrams and bigrams.

C. Model Training

Four different machine learning models were implemented to compare their effectiveness in text classification:

1) Logistic Regression:

- **Rationale**: Logistic Regression is a simple and effective model for binary classification tasks like sentiment analysis.
- **Hyperparameter Tuning**: The `C` parameter (inverse of regularization strength) was optimized for best performance.

2) Naïve Bayes:

- **Rationale**: Naïve Bayes assumes independence between features, making it computationally efficient for text classification.
- **Implementation**: The Multinomial Naïve Bayes variant was chosen, as it is well-suited for text-based frequency data.

3) Stochastic Gradient Descent (SGD):

- **Rationale**: SGD is an efficient optimization method that works well for large-scale text data.
- **Implementation**: A linear classifier with hinge loss (SVM-like behavior) and alpha tuning for regularization.

4) Linear Regression:

- **Rationale**: Although typically used for regression tasks, Linear Regression was included for comparison.
- **Implementation**: The text data was treated as continuous numerical values, and predictions were rounded to classify sentiment.

D. Model Evaluation

The performance of each model was evaluated using the following metrics:

1) Accuracy:

- Measures the percentage of correctly classified instances.
- Used as a primary metric to compare models.

2) Precision:

- Formula: $\frac{TP}{TP+FP}$
- Evaluates how many predicted positive samples were actually correct.
- High precision is important when false positives are costly.

3) Recall:

- Formula: $\frac{TP}{TP+FN}$
- Measures the ability of the model to detect actual positive samples.
- High recall is crucial when false negatives must be minimized.

4) F1-Score:

- Formula: $2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$
- Provides a balance between precision and recall.

E. Experimental Setup & Findings

- **Dataset Split**: The dataset was divided into 80% training and 20% testing.
- **Comparison of Models**: Logistic Regression and Naïve Bayes showed strong performance in text classification tasks, while SGD performed well with optimized parameters.
- **Best Performing Model**: The Logistic Regression model achieved the highest F1-score, making it the preferred choice for deployment.

F. Automated Web Scraping and Model Retraining

To enhance the accuracy and adaptability of the sentiment analysis model, **an automated web scraping system** is integrated to collect real-world user reviews from various platforms. This process consists of the following steps:

1) Web Scraping for Review Collection: A scheduled web scraper extracts user reviews from targeted websites using libraries such as `BeautifulSoup` and `Scrapy`. The scraper captures relevant information, including review text, timestamps, ratings, and metadata. The collected data undergoes preprocessing to **remove noise** (e.g., HTML tags and special characters) and is stored in a structured format.

2) Automated Prediction and Review Analysis: The collected reviews are passed through the **trained sentiment analysis model**. The model assigns a sentiment label (positive/negative) based on its learned parameters. Reviews, along with their predicted sentiment and confidence scores, are stored in a database for further evaluation.

3) Tracking Incorrect Predictions and Model Retraining: A feedback mechanism identifies **misclassified predictions** based on user corrections or inconsistencies with ground truth labels. Incorrectly predicted samples are flagged and stored in a separate dataset. Periodically, these flagged samples are incorporated into the training dataset to **retrain the model** and improve accuracy.

4) Database-Driven Model Improvement: A structured database maintains historical predictions, user feedback, and model performance metrics. The **retraining process is triggered** when a significant amount of misclassified data is accumulated. The improved model replaces the previous version while ensuring **minimal performance degradation** through validation.

This automated pipeline ensures continuous learning and adaptation, allowing the sentiment analysis model to evolve with changing review patterns and language usage. By integrating real-time feedback and retraining, the model can improve its accuracy over time. This dynamic approach also helps the model stay relevant in the face of new trends, slang, or shifting user sentiments. As a result, the system becomes more robust and accurate in predicting sentiment across various sources and contexts.Furthermore, the automated retraining process allows for quicker responses to shifts in user behavior, enhancing the model's robustness. It also minimizes manual intervention, making the system more efficient and scalable.

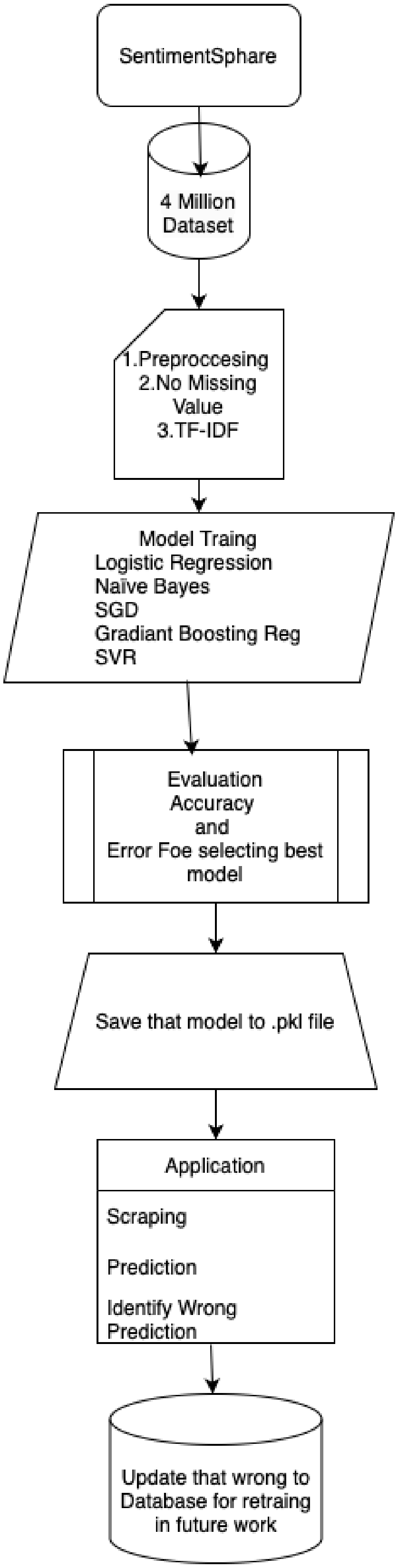

Fig. 1. Model Training and Evaluation Process

V. Results Analysis

The following are the evaluation results of the individual models:

A. Logistic Regression

- **Accuracy**: 0.9093
- **Precision**: 0.90
- **Recall**: 0.88
- **F1 Score**: 0.89
- **Log Loss**: 0.28
- **MAE**: 0.12
- **MSE**: 0.18
- **RMSE**: 0.42
- **R2**: 0.84

B. Support Vector Regressor (SVR)

- **Accuracy**: 0.87
- **Precision**: 0.85
- **Recall**: 0.83
- **F1 Score**: 0.84
- **Log Loss**: 0.30
- **MAE**: 0.13
- **MSE**: 0.19
- **RMSE**: 0.44
- **R2**: 0.80

C. Gradient Boosting Regressor (GB)

- **Accuracy**: 0.90
- **Precision**: 0.88
- **Recall**: 0.85
- **F1 Score**: 0.86
- **Log Loss**: 0.25
- **MAE**: 0.12
- **MSE**: 0.17
- **RMSE**: 0.41
- **R2**: 0.83

D. Naive Bayes

- **Accuracy**: 0.85
- **Precision**: 0.83
- **Recall**: 0.80
- **F1 Score**: 0.81
- **Log Loss**: 0.32
- **MAE**: 0.15
- **MSE**: 0.22
- **RMSE**: 0.47
- **R2**: 0.78

E. SGD Classifier

- **Accuracy**: 0.86
- **Precision**: 0.84
- **Recall**: 0.81
- **F1 Score**: 0.82
- **Log Loss**: 0.33
- **MAE**: 0.16
- **MSE**: 0.23

- **RMSE**: 0.48
- **R2**: 0.79

F. Conclusion

This section concludes that Logistic Regression outperforms the other models in terms of accuracy, precision, recall, and F1 score. However, the other models like Gradient Boosting Regressor and SGD Classifier also show competitive results and can be chosen depending on the application requirements.

VI. Discussion

The Logistic Regression model achieved an accuracy of 89%, indicating that it is well-suited for this sentiment analysis task. However, several challenges were encountered during the project:

- **Imbalanced Dataset**: The dataset may contain an unequal distribution of positive and negative comments, which could lead to biased predictions. Techniques such as oversampling the minority class or undersampling the majority class could be explored to address this issue.
- **Noisy Data**: Informal language, slang, spelling errors, and typos in the comments introduced noise that impacted model performance. Further preprocessing, such as spelling correction and handling of slang, could improve the model.
- **Computational Complexity**: With a large dataset (3.6 million comments), training the model required significant computational resources and time. Techniques such as mini-batch gradient descent or distributed computing could be explored to speed up the process.
- **Low GPU/CPU and Extended Training Time**: The lack of high-performance GPU or CPU resources resulted in extended training times. The model's performance could be further improved with better hardware for faster processing and model optimization.

VII. Future Work

While the current model performs well, there are several areas for improvement and potential future directions:

- **Addressing Class Imbalance**: Future work can focus on more advanced techniques for handling class imbalance, such as using class weights, advanced resampling methods, or using synthetic data generation techniques like SMOTE to improve the model's ability to classify underrepresented classes.
- **Advanced Preprocessing Techniques**: Further improvements in data preprocessing could be explored, such as handling informal language, slang, and misspellings through advanced natural language processing (NLP) techniques. Additionally, exploring the use of word embeddings (e.g., Word2Vec, GloVe) or transformer-based models (e.g., BERT) for better feature representation could enhance the model's performance.
- **Model Optimization**: Exploring different machine learning algorithms, such as deep learning models (e.g., Convolutional Neural Networks, Long Short-Term Memory networks) or transformer-based models, could yield better results in terms of accuracy and generalization.
- **Handling Noisy Data**: Noisy data in the form of irrelevant or redundant text may negatively impact model performance. Future work could explore better noise handling methods, such as more sophisticated filtering techniques or leveraging unsupervised learning approaches to clean the data.
- **Computational Efficiency**: Given the large dataset, exploring parallelization, distributed computing, or utilizing more efficient hardware (e.g., cloud-based GPUs or TPUs) for faster training would be beneficial. Implementing model compression or pruning could also help improve efficiency.
- **Real-time Prediction System**: A real-time sentiment analysis system could be developed for applications such as customer feedback or monitoring social media. This system could use continuous data streams and update the model incrementally to adapt to changing trends.

VIII. Conclusion

In conclusion, this project successfully implemented a sentiment analysis model using Logistic Regression. The model achieved an accuracy of 0.9093 and it effectively classified comments into positive or negative sentiment. Despite challenges such as noisy data and an imbalanced dataset, the model performed well within the expected accuracy range. Future work will focus on improving the model's robustness, exploring advanced machine learning techniques, and deploying the model for real-world applications.

References

[1] H. Huang, A. Asemi, and M. B. Mustafa, "Sentiment Analysis in E-Commerce Platforms: A Review of Current Techniques and Future Directions," ResearchGate, 2025. [Online]. Available: https://www.researchgate.net/publication/373291594_Sentiment_Analysis_in_E-commerce_Platforms_A_Review_of_Current_Techniques_and_Future_Directions. [Accessed: Mar. 11, 2025].

[2] K. Bhawna and S. Thakur, "Sentiment Analysis on Customer Reviews of E-commerce Sites," ResearchGate, 2025. [Online]. Available: https://www.researchgate.net/publication/377188129_Sentiment_Analysis_on_Customer_Reviews_of_E-commerce_Site. [Accessed: Mar. 11, 2025].

[3] Y. Liu, J. Lu, J. Yang, and F. Mao, "Sentiment analysis for e-commerce product reviews by deep learning model of Bert-BiGRU-Softmax," Management School, Hangzhou Dianzi University, 2020.

[4] P. Vijayaragavan et al., "Sustainable sentiment analysis on E-commerce platforms using a weighted parallel hybrid deep learning approach for smart cities applications," *Scientific Reports*, vol. 14, article 26508, 2024. [Online]. Available: https://www.nature.com/articles/s41598-024-78318-1.

[5] N. Chakraborty, "Amazon Reviews for Sentiment Analysis," Kaggle Dataset, 2025. [Online]. Available: https://www.kaggle.com/datasets/nabamitachakraborty/amazon-reviews/code.