Enhanced Sentiment Analysis Of App Reviews Using Naïve Bayes And Ensemble Learning

V. Dhanakoti, Anitha Varshini. A, Athmanathan K, Balineni Akhila
SRM Valliammai Engineering College

Abstract: **Sentiment analysis is essential for understanding user opinions and enhancing decision-making across industries. This project develops a sentiment classification system using Naïve Bayes algorithms—Multinomial Naïve Bayes (MNB), Bernoulli Naïve Bayes (BNB), Complement Naïve Bayes (CNB), and an Ensemble Voting Classifier—to analyze Google Play Store app reviews. The system efficiently classifies reviews into positive or negative sentiments through extensive text preprocessing, including stopword removal, special character filtering, and TF-IDF vectorization for improved feature extraction. The preprocessed data is used to train and test Naïve Bayes models, with performance evaluated using accuracy, confusion matrices, precision-recall curves, and ROC-AUC scores. Among the models, ComplementNB outperforms others with an AUC of 0.92, demonstrating a strong balance between precision and recall, whereas BernoulliNB struggles due to a high false negative rate. To enhance usability, a Flask-based web application is integrated, allowing users to input reviews and visualize classification results in real time. The study underscores the effectiveness of Bayesian machine learning models for sentiment analysis, demonstrating their scalability and reliability in text classification tasks. The proposed system provides a practical and efficient solution for sentiment analysis in app reviews, aiding businesses in decision-making and user experience improvements.**

Index Terms – Sentiment Analysis, Machine Learning, ComplementNB, BenouliNB, AUC Score, TF-IDF Vectorization.

I. INTRODUCTION

Sentiment analysis, a crucial subfield of Natural Language Processing (NLP), has gained significant traction due to the exponential growth of online social networks and electronic media-based societies [1]. Organizations actively assess public opinion on products and services by analyzing social media data, including blogs, online forums, tweets, and product reviews [2]. This assessment plays a vital role in decision-making, service enhancement, and product quality improvement [3]. The increasing reliance on digital platforms for business operations has underscored the impact of opinionated web text on socio-economic systems [4]. Sentiment analysis applications extend across diverse domains, including election campaigns, healthcare monitoring, consumer product reviews, and public awareness services [5]. With the advancement of computational power and machine learning techniques, sentiment analysis has evolved into a sophisticated text classification task [6]. Traditional sentiment analysis methods classify text at three levels: document, sentence, and word level, while recent advancements employ deep convolutional neural networks for character-level feature extraction [7].

Sentiment analysis enables companies to categorize user comments into queries, complaints, suggestions, opinions, or product endorsements, streamlining customer engagement and prioritization [8]. Additionally, it aids in filtering spam content, refining recommendation systems, and enhancing collaborative filtering-based applications. NLP serves as the foundation of sentiment analysis, allowing machines to understand and derive insights from textual data, facilitating virtual assistants, query resolution, and content summarization. Sentiment analysis approaches fall into two broad categories: rule-based and automated machine learning-based methods. Rule-based methods rely on predefined word lists to classify sentiments, whereas machine learning-based techniques, such as Support Vector Machines (SVM), Linear Regression, and Recurrent Neural Networks (RNN), provide deeper contextual analysis. As sentiment analysis continues to evolve, it remains a fundamental tool for businesses and researchers in understanding user opinions and improving decision-making.

II. LITERATURE SURVEY

Sentiment analysis has gained significant traction as an essential aspect of Natural Language Processing (NLP), with researchers exploring various methodologies to enhance its accuracy and effectiveness. Saxena et al. [1] provided a

comprehensive introduction to sentiment analysis, discussing fundamental concepts, evaluation metrics, tools, challenges, and applications. Their work highlighted how sentiment analysis plays a crucial role in decision-making across industries, utilizing various machine learning techniques to classify sentiments. Carter et al. [2] focused on applying NLP to assess the emotional well-being of student-athletes during the COVID-19 pandemic. They analyzed social media content to identify patterns of distress and emotional fluctuations, demonstrating the importance of sentiment analysis in healthcare and psychology.

Chandrasekaran et al. [3] examined sentiment trends on Twitter related to COVID-19, using temporal analysis to track changes in public perception. Their study emphasized the impact of social media sentiment on policymaking and crisis management. Similarly, Samuel et al. [4] used machine learning techniques to classify tweets about COVID-19, providing insights into public sentiment variations during different pandemic phases. Their work showcased how sentiment analysis can be leveraged for real-time monitoring of public opinion. Iyer and Kumaresh [5] extended this research by analyzing Twitter sentiment on the coronavirus outbreak, applying machine learning algorithms to identify dominant emotions in public discourse. Their study reinforced the significance of sentiment analysis in understanding public concerns and information dissemination trends during health crises.

Singh et al. [6] explored sentiment analysis of COVID-19-related tweets using machine learning techniques. They compared various classifiers, demonstrating that advanced machine learning models could outperform traditional sentiment classification methods. Aziz and Dimililer [7] introduced an ensemble-weighted majority vote classifier for Twitter sentiment analysis, improving classification accuracy by combining multiple learning models. Their work contributed to enhancing sentiment classification robustness and reliability. Riadi et al. [8] examined mobile forensics for cyberbullying detection using Term Frequency-Inverse Document Frequency (TF-IDF). Their study illustrated how sentiment analysis techniques could be extended to cybersecurity applications, identifying harmful content on mobile platforms.

Raheja and Munjal [9] took a different approach by classifying Microsoft Office vulnerabilities, aiming to enhance software security using sentiment analysis principles. Their research demonstrated how sentiment classification could aid in detecting and mitigating software security threats. Tenasv et al. [10] applied sentiment analysis to Bangla song reviews, introducing a lexicon-based backtracking approach to improve accuracy. Their work highlighted the versatility of sentiment analysis across different languages and cultural contexts.

III. MATERIALS AND METHODS

A) Proposed Work:

The proposed sentiment analysis system utilizes Naïve Bayes classification algorithms to efficiently classify Google Play Store reviews as positive or negative. It comprises key components such as data preprocessing, feature extraction, model training, evaluation, and deployment via a Flask-based web application. The system optimizes classification accuracy while maintaining computational efficiency for large-scale text data. Preprocessing includes lowercasing, stopword removal, tokenization, and TF-IDF vectorization, followed by Chi-Square feature selection. Three Naïve Bayes models—MultinomialNB, BernoulliNB, and ComplementNB—are employed, with an Ensemble Voting Classifier enhancing accuracy. ComplementNB achieves the highest AUC (0.92). The final model is deployed via Flask, enabling real-time sentiment classification and visualization, offering a scalable, automated solution for sentiment analysis in user reviews.

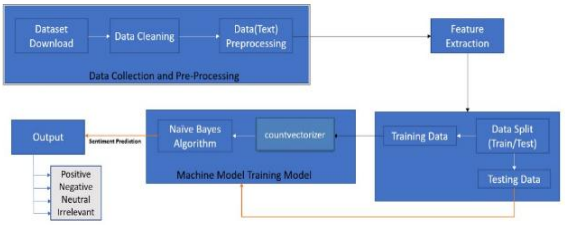

Fig.1 Block Diagram

The Block Diagram (fig.1) for sentiment analysis follows a structured pipeline, starting with Data Collection and Pre-Processing. This phase includes dataset downloading, data cleaning, and text preprocessing (tokenization, stopword removal, and normalization). Feature Extraction is then performed using CountVectorizer, converting text into numerical features. The Machine Model Training Model employs a Naïve Bayes Algorithm, trained on

labeled sentiment data. The dataset is split into training and testing sets, ensuring proper evaluation. The trained model classifies reviews into Positive, Negative, Neutral, or Irrelevant sentiments. Finally, the system provides real-time Sentiment Prediction as output. This architecture ensures an efficient and scalable machine learning model, making it suitable for sentiment analysis of textual reviews in applications like user feedback analysis.

B) Dataset Collection:

The dataset for this sentiment analysis system is collected from Kaggle, containing Google Play Store app reviews. It consists of three columns: package_name (identifying the app), review (user-written text), and polarity (sentiment label, where 0 = negative and 1 = positive). For instance, users expressed dissatisfaction with the Facebook app, mentioning issues like the lack of an "offline" mode, message notification problems, and feature malfunctions. These reviews provide valuable insights into user sentiment, aiding in the development of an efficient classification model.

C) Pre-Processing:

Pre-processing involves cleaning and transforming raw text data by tokenization, stopword removal, and special character filtering to prepare it for machine learning.

i) Data Preprocessing: Preprocessing involves cleaning and transforming raw text data for machine learning. A tokenizer is applied to split text into words using a regular expression ([a-zA-Z0-9]+), ensuring only meaningful words remain. This process helps in handling noise like special characters. The cleaned data is then split into training and testing sets, where the training set is used to build the model, and the testing set evaluates its performance. Proper preprocessing ensures better feature extraction and enhances classification accuracy.

ii) Feature Extraction: CountVectorizer is used to convert text into numerical vectors by counting word occurrences. Each document is represented as a word frequency vector, making it suitable for machine learning. Parameters like stop_words (removes common words), ngram_range (defines word sequences), and custom tokenizers help refine feature extraction. The fit_transform method generates a sparse matrix, where each row is a document and columns represent words. This matrix serves as input for sentiment analysis, improving classification performance.

D) Training & Testing:

The dataset is divided into training and testing sets to evaluate the model's performance. Typically, 80% of the data is used for training, where the machine learning model learns patterns from labeled data, and 20% is used for testing, ensuring the model generalizes well to unseen reviews. The Naïve Bayes classifiers (MultinomialNB, BernoulliNB, ComplementNB) are trained using TF-IDF feature vectors extracted from text data. The trained models are then tested on unseen data to measure accuracy, precision, recall, and AUC scores. The best-performing model is selected based on evaluation metrics for sentiment classification in real-world applications.

E) Algorithms:

MultinomialNB: Effective for text classification, it models word frequency distributions and is well-suited for sentiment analysis. It assumes feature independence and works well with TF-IDF vectorized data, making it useful for distinguishing between positive and negative reviews.

The equation for Multinomial Naïve Bayes is:

$$P\ (C_k|X) = \frac{P(C_k)\prod_{i=1}^{n} P(X_i|C_k)}{P(X)}\ (1)$$

Where:

- $P(C_k|X)$is the posterior probability of class C_k given the feature vector X.
- $P(C_k)$ is the prior probability of class C_k.
- $P(X_i|C_k)$ is the likelihood of feature X_i given class C_k, computed using word counts.
- $P(X)$ is the evidence (normalization factor), which ensures probabilities sum to 1.

BernoulliNB: Designed for binary features, it operates on presence or absence of words rather than frequency. This makes it effective when text data is converted into a binary representation, enhancing classification performance in cases with strong word presence indicators.

Since BernoulliNB deals with binary features (Xi∈{0,1}X_i \in \{0,1\}, the likelihood is computed as:

$$P(X\_i \mid C\_k) = p\_\{i,k\}\^\{X\_i\} (1 - p\_\{i,k\})\^\{(1 - X\_i)\} (2)$$

Where:

- Ni, k is the number of documents in class Ck containing word iii.
- Nk is the total number of documents in class CkC_kCk.

- α is the Laplace smoothing parameter (to prevent zero probabilities).

ComplementNB: Optimized for imbalanced datasets, it improves classification accuracy by adjusting weights for underrepresented classes. It outperforms other variants by reducing bias in sentiment classification.

$$P(X\_i \mid C\_k) = \frac{N\_\{i,\neg k\} + \alpha}{N\_\{\neg k\} + \alpha d} \quad (3)$$

Where:

- Ni,¬k = The count of feature Xi in all classes except Ck.
- N¬k = The total count of all features in all classes except Ck.
- d = The total number of features (vocabulary size in text classification).
- α = Laplace smoothing parameter.

Ensemble Voting: Combines predictions from multiple models to enhance accuracy and robustness. By aggregating outputs from Naïve Bayes classifiers, it mitigates individual model weaknesses, leading to more reliable sentiment predictions.

IV. EXPERIMENTAL RESULTS

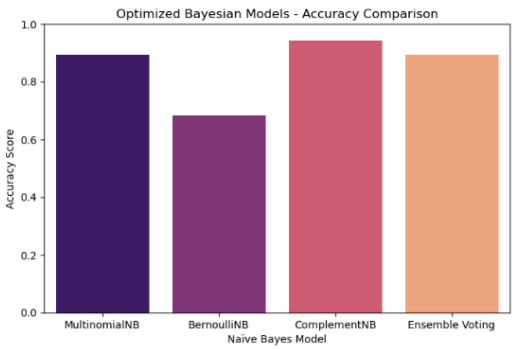

Fig.2 Accuracy Comparison Graph

Accuracy: The accuracy of a test is its ability to differentiate the patient and healthy cases correctly. To estimate the accuracy of a test, we should calculate the proportion of true positive and true negative in all evaluated cases. Mathematically, this can be stated as:

$$Accuracy = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FP} + \mathrm{TN} + \mathrm{FN}} \quad (4)$$

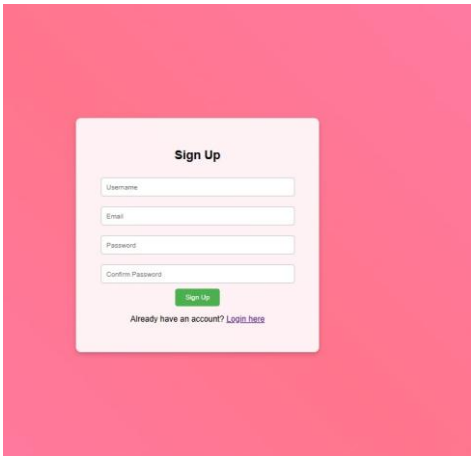

Fig.3 Signup Page

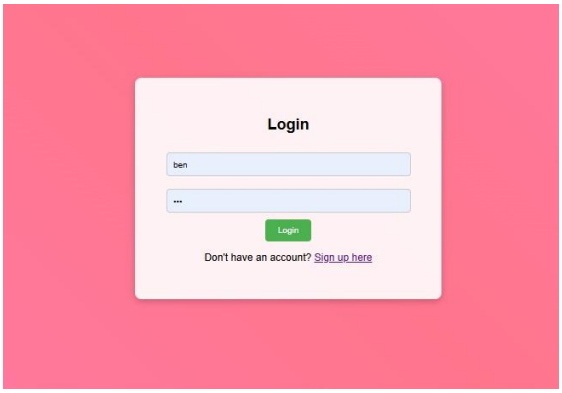

Fig.4 Login Page

Fig.5 Upload Input Page

Fig.6 Final Outcome

V. CONCLUSION

Evaluating multiple Bayesian classification models, including MultinomialNB, BernoulliNB, ComplementNB, and an Ensemble Voting Classifier, provided valuable insights into their effectiveness for classification tasks. Analysis using confusion matrices, ROC curves, and precision-recall curves highlighted the strengths and limitations of each model. ComplementNB demonstrated the highest true positive rate and the lowest false negatives, ensuring a strong balance between precision and recall, making it highly effective for applications such as sentiment analysis, fraud detection, and spam filtering. Despite good performance from MultinomialNB and the Ensemble Voting Classifier, BernoulliNB exhibited significant weaknesses due to its high false negative rate, making it unsuitable for recall-sensitive tasks. These findings emphasize the necessity of choosing the right classifier based on dataset characteristics and specific application needs. Optimizing classifier selection through machine learning techniques enhances classification accuracy and scalability, leading to more efficient solutions. A well-chosen model improves decision-making, particularly in large-scale text processing scenarios. The results contribute to better understanding of Bayesian classifiers, enabling informed choices for real-world classification challenges requiring reliability and precision.

Future Scope can enhance classification accuracy by integrating deep learning models like LSTMs and transformers for contextual understanding. Expanding the dataset with diverse linguistic patterns can improve generalization. Optimizing feature extraction techniques, such as word embeddings, can further refine sentiment classification. Deploying the model in real-time applications with adaptive learning will enhance scalability. Incorporating multilingual support and handling sarcasm detection will broaden applicability across various domains.

REFERENCES

[1] Akrati Saxena, Harita Reddy, Pratishtha Saxena, "Introduction to Sentiment Analysis Covering Basics, Tools, Evaluation Metrics, Challenges, and Applications", Principles of Social Networking, vol.246, pp.249, 2022.

[2] F. Carter, S. Gulavani, D. James, C. H. Amy and P. Jason, "A Tale of Two Cities: COVID-19 and the Emotional Well-Being of Student-Athletes Using Natural Language Processing", Frontiers in Sports and Active Living, vol. 3, pp. 227-233, 2021.

[3] R. Chandrasekaran, V. Mehta, T. Valkunde and E. Moustakas, "Twitter talk on COVID-19: A temporal examination of topics trends and sentiments", J Med Internet Res, vol. 22, no. 10, pp. e22624, 2020.

[4] J. Samuel, G. G. M. N. Ali, M. M. Rahman, E. Esawi and Y. Samuel, "COVID-19 public sentiment insights and machine learning for tweets classification", Information, vol. 11, no. 6, pp. 1-22, 2020.

[5] P. Iyer and S. Kumaresh, "Twitter Sentiment Analysis On Coronavirus Outbreak Using Machine Learning Algorithms", European Journal of Molecular & Clinical Medicine, vol. 7, no. 3, pp. 2663-2676, 2020.

[6] S. K. Singh, P. Verma and P. Kumar, "Sentiment analysis of Covid -19 epidemic using Machine Learning Algorithm on Twitter", Journal of Critical Reviews, vol. 7, no. 18, pp. 2565-2572, 2020.

[7] R. H. H. Aziz and N. Dimililer, "Twitter Sentiment Analysis using an Ensemble Weighted Majority Vote Classifier", 2020 International Conference on Advanced Science and Engineering (ICOASE), pp. 103-109, 2020.

[8] I. Riadi, S. Sunardi and P. Widiandana, "Mobile Forensics for Cyberbullying Detection using Term Frequency - Inverse Document Frequency (TF-IDF)", J. Ilm. Tek. Elektro Komput. dan Inform, vol. 5, pp. 68-76, 2019.

[9] Raheja Supriya and Geetika Munjal, "Classification of Microsoft office vulnerabilities: a step ahead for secure software development", Bio-inspired Neurocomputing, pp. 381-402, 2021.

[10] Rabey a Tenasv, Narayan Ranjan Chakraborty, Saniida Ferdous, Manoranjan Dash and Ahmed Al Marouf, "Sentiment analysis of bangla song review-a lexicon based backtracking approach", 2019 IEEE International Conference on Electrical Computer and Communication Technologies (ICECCT), pp. 1-7, 2019.