

Sentiment Analysis of Apple Product Reviews: Decoding Brand Loyalty

CSCI 6401 - Data Mining

**Team: Data Ops**

**Team Members:**

* Team Lead & Data Scientist: Naga Swetha [Nallaninnall10@unh.newhaven.edu](mailto:Nallaninnall10@unh.newhaven.edu)
* Data Engineer: Govardhan Namala – [gnama2@unh.newhaven.edu](mailto:gnama2@unh.newhaven.edu)
* Data analyst: Amani Ponnam – [aponn6@unh.newhaven.edu](mailto:aponn6@unh.newhaven.edu)

# Abstract

We analyzed how consumer sentiments in product reviews impact brand loyalty and perception in Apple's product ecosystem. We used the Apple Product Review dataset from Kaggle to analyze sentiment patterns through data mining techniques. Our approach combined traditional models like Logistic Regression and Naive Bayes with advanced techniques such as XGBoost, ensemble methods, and CNNs. By optimizing hyperparameters with GridSearchCV and RandomizedSearchCV, we improved model performance significantly. The Enhanced CNN achieved the best results, with an accuracy of 88.56% and an AUC of 0.97, demonstrating its ability to handle complex sentiment patterns. While traditional models provided baseline insights, advanced deep learning approaches delivered actionable analytics for understanding consumer sentiment. Future research can expand these methods to other domains and incorporate the latest techniques for even better results.

# 

# 

# INTRODUCTION

Apple's unmatched brand loyalty stems from its innovative products and strong customer relationships. Understanding consumer sentiments offers actionable insights to improve marketing strategies and enhance customer satisfaction. This study analyzes sentiment patterns in product reviews to explore their impact on Apple's brand perception. We applied machine learning and deep learning techniques to effectively classify and interpret these sentiments.

# Related Work

1. **Tyagi & Sharma (2017):**Tyagi and Sharma applied Support Vector Machine (SVM) to handle ambiguous sentiments effectively using TF-IDF, achieving an accuracy of 89.98%. Their work demonstrated the model's ability to classify complex text data. However, limited information on hyperparameter tuning highlighted the potential for further optimization, which could improve SVM performance.
2. **Kubrushy et al. (2022)**Kubrushy and colleagues utilized XGBoost and Gradient Boosting methods to ensure stability in tree-based classification tasks, achieving an accuracy of 89.87%. Despite their effectiveness, these models faced challenges with overfitting when using a high number of trees, which restricted their ability to generalize to new data.
3. **Su et al. (2013)**Su and collaborators employed ensemble stacking techniques to improve classification performance by leveraging diversity among classifiers and opinion summaries. While their approach demonstrated enhanced accuracy, it primarily focused on majority voting and did not incorporate modern deep learning approaches like BERT or LSTM, limiting its relevance in current research contexts.
4. **Kim (2014)**Kim demonstrated the effectiveness of Convolutional Neural Networks (CNNs) in capturing nuanced textual patterns for sentiment analysis, achieving an accuracy of 88.89%. While CNNs performed well, they struggled with context-rich phrases and required significant computational resources, presenting a limitation in practical applications.
5. **Devlin et al. (2018)**Devlin and colleagues introduced BERT, which provided state-of-the-art contextual understanding for text classification tasks. While BERT delivered high accuracy across various tasks, its computational intensity and need for significant domain-specific fine-tuning posed challenges for widespread applications.

# Proposed Method

### Data Acquisition and Preprocessing:

We used the **Apple Product Review Dataset** from [Kaggle](https://www.kaggle.com/datasets/mslatifpour/apple-product-review-dataset), which contains textual reviews labeled as positive, negative, or neutral. The dataset was preprocessed to make it suitable for machine learning models. Key steps included:

1. **Removing Punctuation and HTML Tags**: Unnecessary elements like special characters and HTML tags were removed to ensure data cleanliness.
2. **Stopword Removal**: Commonly used words such as "the" and "is" were excluded as they provided little to no value in sentiment classification.
3. **Tokenization**: Reviews were split into individual words to facilitate text processing.
4. **TF-IDF Vectorization**: Term Frequency-Inverse Document Frequency (TF-IDF) was employed to convert textual data into numerical features while emphasizing significant words.
5. **Data Splitting**: To evaluate model performance effectively, the dataset was divided into 80% for training and 20% for testing

### Models Implemented

1. **Logistic Regression**: Used as a baseline model to evaluate linear separability in the data.
2. **Naive Bayes**: Chosen for its probabilistic principles, which work effectively for text-based classification tasks.
3. **SVM (Support Vector Machine)**: Leveraged for handling high-dimensional data with kernel tricks for better performance.
4. **XGBoost**: An ensemble method that uses gradient boosting for improved accuracy and robustness.
5. **Ensemble Model**: Combined Logistic Regression and Naive Bayes through soft voting to leverage their complementary strengths.
6. **CNN (Convolutional Neural Network)**: Designed to capture nuanced sentiment patterns in textual data using embedding layers, convolutional filters, and pooling layers.

### Evaluation Metrics

The models were evaluated using the following metrics:

* **Accuracy**: The proportion of correctly predicted sentiments.
* **AUC (Area Under the Curve)**: A measure of the ability of the model to distinguish between classes.
* **Precision, Recall, and F1-Score**: Metrics used to assess performance across all sentiment classes.  
  We visualized the results using **ROC curves** and **confusion matrices** to better understand model performance.

# Results

The Enhanced CNN achieved the highest performance with an **accuracy of 88.56%** and an **AUC of 0.97**, outperforming all other models.

* **Baseline Models**: Logistic Regression and Naive Bayes achieved moderate accuracies of 64% and 68%, respectively.
* **Advanced Models**: SVM and XGBoost provided better results, but their performance was limited by overfitting issues and parameter sensitivity.
* **Ensemble Models**: By combining Logistic Regression and Naive Bayes, the ensemble model improved classification performance, addressing some weaknesses of individual models.

Visualization techniques such as confusion matrices demonstrated the Enhanced CNN's reduced misclassification rates, especially in neutral and slightly negative sentiments.

# Discussion

The results showed that traditional machine learning models, while computationally efficient, struggled to capture the complexities of consumer sentiments. Ensemble methods helped to mitigate these limitations by combining the strengths of individual classifiers. The Enhanced CNN, however, provided the most robust performance due to its ability to learn nuanced patterns in textual data. Despite these successes, challenges such as high computational costs and difficulties in handling sarcasm or mixed sentiments were noted.

# Appendix

The project's source code and related documentation are accessible on the GitHub repository:  
[GitHub Link](https://github.com/NallaniSwetha345/CSCI-6401-01_DataOps_Project).

# 

# 

# References

1. Tyagi, E., & Sharma, A. K. (2017). Sentiment analysis of product reviews using support vector machine learning algorithm. *Indian Journal of Science and Technology, 10*(35), 1–8. <https://doi.org/10.17485/ijst/2017/v10i35/118965>
2. Kubrusly, J., Neves, A. L., & Marques, T. L. (2022). A statistical analysis of textual e-commerce reviews using tree-based methods. *Open Journal of Statistics, 12*(3), 357–372.<https://doi.org/10.4236/ojs.2022.123023>
3. Su, Y., Zhang, Y., Ji, D., Wang, Y., & Wu, H. (2013). Ensemble learning for sentiment classification. *Lecture Notes in Artificial Intelligence, 7717*, 84–93. <https://doi.org/10.1007/978-3-642-36338-2_10>
4. Kim, Y. (2014). Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.<https://arxiv.org/pdf/1408.5882>
5. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.<https://arxiv.org/pdf/1810.04805>
6. Kaggle. Apple Product Review Dataset. Retrieved from<https://www.kaggle.com/datasets/mslatifpour/apple-product-review-dataset>