University of New Haven

Team: Data Ops

Team Members:

- Team Lead & Data Scientist: Naga Swetha Nallaninnall10@unh.newhaven.edu
- Data Engineer: Govardhan Namala – gnama2@unh.newhaven.edu
- Data analyst: Amani Ponnam – aponn6@unh.newhaven.edu

Research Question:

How do consumer sentiments in product reviews influence brand loyalty and perception of Apple's product ecosystem?

The Dataset we chose is the Apple Product Review Dataset which is available in Kaggle at this link

Github Repository:

A GitHub repository has been created for this final project and all the source code used for this phase -5 is uploaded in a different branch named Phase-5

You can access the repository at this link. The code can be seen in the said branch.

Data Mining Techniques and Methodologies:

We employed a step-by-step methodology, initially leveraging traditional data mining techniques to create baseline models. These included:

1. **Logistic Regression**: chosen for its interpretability and straightforward implementation.
2. **Naive Bayes**: applied due to its efficiency with text-based data and ability to classify by word frequency.
3. **Support Vector Machine (SVM)**: utilized to capture high-dimensional patterns in text data, despite the need for extensive tuning.
4. **XGBoost**: selected for its ensemble nature, capable of handling complex relationships within the data.

Our early results highlighted moderate predictive accuracy across models, with Naive Bayes achieving 68% accuracy and an AUC of 0.45, while Logistic Regression offered 64% accuracy and an AUC of 0.34. Both models showed some ability to distinguish sentiment patterns, but SVM and XGBoost delivered suboptimal results at 59% and 65% accuracy, respectively. These outcomes suggested that the existing data and feature extraction methods captured sentiment nuances insufficiently for high predictive power.

Key hyperparameters were selected and tuned for each model:

1) Logistic Regression: C=1.0, solver='liblinear'
2) Naive Bayes: alpha=1.0
3) SVM: kernel='rbf', C=2.0, gamma=0.1
4) XGBoost: learning_rate=0.1, max_depth=6, n_estimators=100

All experiments were conducted on an I7 processor with 512 GB SSD and 16 GB RAM, implemented within a Jupyter Notebook environment.

Given the moderate success of Naive Bayes and Logistic Regression, we hypothesized that combining these models in an Ensemble Model would amplify their strengths. Naive Bayes proved effective with text frequency features, while Logistic Regression contributed robustness to the decision boundary. By applying soft voting, this ensemble model reached an improved accuracy outperforming the individual models. Our most significant advancement, however, came with the deployment of a Convolutional Neural Network (CNN). By implementing an optimized CNN architecture with a carefully tuned Adam optimizer, dropout layers, and additional convolutional filters, we achieved significant gains. Each of these model's configuration will be discussed further.

Metrics:

Each model was evaluated based on the following metrics:

Accuracy: Proportion of correct predictions.

Precision, Recall, and F1-Score: To understand how well each model performed on each sentiment category.

AUC (Area Under the ROC Curve): To evaluate how well the model distinguishes between positive and negative sentiments.

Outcomes of Data Mining Techniques Using Varied Performance Metrics:

1) **Logistic Regression Model**
- Configuration: solver=liblinear, C=1.0
- Purpose: We initially used Logistic Regression as a baseline model due to its simplicity and interpretability.
- Performance: Achieved an accuracy of 64% with an AUC of 0.34. Precision and recall were moderate, indicating that while the model captured some sentiment patterns, it lacked sensitivity in distinguishing between sentiment categories.

2) **Multinomial Naive Bayes**
- Configuration: Optimized alpha parameter using GridSearchCV
- Purpose: Given Naive Bayes' efficiency with text data, we anticipated strong performance, especially for frequency-based feature extraction.
- Performance: Naive Bayes provided the best results among traditional models, with an accuracy of 68% and an AUC of 0.45. Its higher recall and precision confirmed it as the most suitable traditional model for text classification in this dataset.

3) **Support Vector Machine (SVM)**
- Configuration: kernel=rbf, C=2.0, gamma=0.1
- Purpose: We implemented SVM to leverage its robustness in handling high-dimensional text data, with the expectation that it could better capture subtle sentiment features.
- Performance: SVM achieved 59% accuracy with an AUC of 0.32, indicating that it struggled with this dataset. Precision, recall, and F1-score metrics reflected limited discrimination ability, likely due to the model's sensitivity to parameter tuning and the complexity of sentiment nuances.

4) **XGBoost Classifier**
- Configuration: learning_rate=0.1, max_depth=6, n_estimators=100
- Purpose: We chose XGBoost to handle complex relationships in data through gradient boosting, hoping it would improve predictive power.
- Performance: XGBoost performed slightly better than SVM, reaching 65% accuracy with an AUC of 0.35. Precision and recall were similar to those of Logistic Regression, indicating moderate performance, but the computational requirements were considerably higher.

5) **Ensemble Model (Naive Bayes + Logistic Regression)**
- Configuration: Soft voting, combining Naive Bayes and Logistic Regression
- Purpose: Combining the strengths of Naive Bayes (effective with text) and Logistic Regression (interpretability) aimed to improve performance.
- Performance: This ensemble approach boosted accuracy to 85.95% when compared to previous models

6) **Ensemble Model with Tuned Logistic Regression**

- Configuration: GridSearchCV-tuned Logistic Regression for C values and liblinear solver, combined with Naive Bayes
- Purpose: By refining Logistic Regression's regularization, this model sought to enhance the ensemble's stability.
- Performance: This model showed a improvement over first ensemble model with an accuracy of 87.91%.

7) **Initial CNN Model**

- Architecture: Embedding layer (128 dimensions) → Conv1D (128 filters, kernel size 5, ReLU) → GlobalMaxPooling → Dense (64 units, ReLU) → Dropout (0.5) → Dense output layer (softmax for 3 classes)
- Optimizer: Adam with default parameters
- Performance: The initial CNN model achieved 87.25 % accuracy, marking a considerable improvement over traditional models but falls short on Ensemble models.

8) **Enhanced CNN Model with Modified Adam Optimizer**

- Architecture: Embedding layer → SpatialDropout1D (0.3) → Conv1D (128 filters, kernel size 5, ReLU, L2 regularization) → Additional Conv1D (128 filters, kernel size 3, ReLU, L2 regularization) → GlobalMaxPooling → Dense (128 units, L2 regularization) → Dropout (0.6) → Dense output layer (softmax for 3 classes)
- Optimizer: Adam with a reduced learning rate of 0.0001
- Performance: This optimized CNN, trained with early stopping, reached 88.56% accuracy with high precision, recall, and F1-score(89) across all classes. The AUC score rose to 0.97, confirming this model's superiority in recognizing subtle sentiment patterns.

Visualization Techniques

To communicate our findings effectively, we employed several visualization techniques:

- **ROC Curves**: Generated for each model, allowing us to compare their ability to distinguish between positive, negative, and neutral sentiments. The ROC curve for the enhanced CNN model showed a significant improvement, indicating a better balance between sensitivity and specificity.
- **Confusion Matrices**: Displayed the distribution of correctly and incorrectly classified sentiments for each model. The CNN model's confusion matrix showed fewer misclassifications, particularly in distinguishing between neutral and negative sentiments, compared to other models.
- **Consolidated Performance Table**: Summarized all metrics for each model, providing a clear side-by-side comparison that highlighted the CNN's superior performance. This table allowed for quick evaluation of each model's effectiveness in meeting the project's objectives.
- Histogram for all Accuracies: Summarized all accuracies for each model and visulaised in a histogram.

Conclusion:

In our exploration of how consumer sentiments influence perceptions of Apple's brand, we found that traditional models offered only moderate predictive capability. Although the Naive Bayes model initially performed best with an accuracy of 68%, it became evident that sentiment patterns in the dataset lacked the distinctiveness necessary for high classification accuracy.

The ensemble approach, combining Naive Bayes and Logistic Regression, provided some improvement, yet the overall predictive power remained limited. It was only when we implemented advanced deep learning architectures, specifically our optimized CNN model, that we saw a substantial leap in performance, achieving a maximum accuracy of 82%. This breakthrough underlined the complexity of sentiment nuances and the limitations of simpler models in capturing such subtleties.

Our findings suggest that while traditional models can provide foundational insights, deep learning methods like CNNs are better suited to the intricate nature of sentiment analysis in product reviews. Future research could build upon these results by implementing even more advanced NLP techniques, such as transformer-based models, which may yield deeper insights given appropriate computational resources.