Lord Charité Igirimbabazi
COSC74, Machine Learning and Statistical Analysis
Dartmouth College, Fall 2024

Amazon Review Classification: Project Write-Up

1. Objective

The goal of this project was to build a binary classification model to predict whether an Amazon review is a high-star (1) or low-star (0) review. The dataset provided includes several features such as reviewText, summary, product-related information (verified, vote, category), and more. These features were used to develop a model capable of accurately classifying reviews.

2. Data Preprocessing and Feature Engineering

- Handling Missing Data:
    - For textual columns ("reviewText", "summary", "category"), missing values were replaced with empty strings. This approach ensures no data is lost, allowing all reviews to be processed.
    - For categorical columns ("verified", "vote", "category"), missing values were handled by converting them into strings to ensure compatibility with machine learning models.

- Feature Engineering:
    - Textual Features: Combined the "reviewText" and "summary" into a new feature called "combined_text" to capture a richer representation of each review.
    - Categorical Features: Included additional features such as "verified", "vote", and "category". These features provide important context about the review (e.g., verified status, helpful votes, product category).
    - Combined Features: Concatenated the "reviewText", "summary", "verified", "vote", and "category" into a single text field for each review. This combined feature served as the primary input for the model.

- Text Processing:
    - The "combined_text" feature was processed using the TF-IDF Vectorizer to convert text data into numerical features.
    - The TF-IDF approach helps to down-weight common stop words and emphasize informative terms that occur less frequently.

3. Model Selection and Hyperparameter Tuning

Three machine learning models were selected for evaluation: Logistic Regression, Perceptron, and Random Forest. CalibratedClassifierCV to improve the probability calibration and class_weight='balanced' to handle class imbalance.

1. Logistic Regression: Chosen for its simplicity and interpretability
2. Perceptron: Chosen for its ability to handle large datasets effectively.
3. Random Forest: Chosen for its ability to model complex relationships between features.

4. Model Training and Evaluation

The dataset was split into training and validation sets (80/20 split). In addition, a cross-validation (5-fold) was used to evaluate model performance, ensuring robust and unbiased results.

The following evaluation metrics were used for performance calculations:

- Accuracy: The proportion of correct predictions.
- F1 Macro Score: The average of the F1 scores for both classes, treating both high-star and low-star reviews equally.
- Recall Macro Score: The average recall score for both classes, measuring the model's ability to correctly identify high and low-star reviews.
- ROC AUC Score: A metric that evaluates the model's ability to distinguish between the two classes.

5. Model Performance

Model	Accuracy (%)	F1 Macro Score (%)	Recall Macro Score (%)	ROC AUC Score (%)
Logistic Regression	**85.78**	**84.96**	**84.60**	**92.79**
Perceptron	**81.70**	**80.42**	**79.82**	**89.95**
Random Forest	**84.62**	**83.39**	**82.52**	**92.20**

➔ Best Model: Logistic Regression performed the best consistently based on Accuracy, F1 Macro Score, and ROC AUC Score, making it the final model selected for predictions on the test set. Therefore, Logistic regression was chosen as the final model based on its superior performance in evaluation metrics.

The Logistic Regression model was applied to the test set, and the predictions were saved in the required "test_predications.csv" format for submission.