Sentiment Analysis

In this supervised machine learning task, I performed binary sentiment analysis using various methods. Below is a brief summary of my approach:

**Data Discovery:**

- I began by exploring the dataset and observed an initial data imbalance issue. In this case of slight dataset imbalance, where one class has a minor numerical deficit compared to another (e.g., 100 vs. 140 samples), undersampling the majority class isn’t necessary as it reduces data. Instead, I use balanced accuracy scoring during hypeparameters tuning.

**Data Preprocessing**:

In the preprocessing phase, I implemented several crucial steps:

- Removal of Numbers

- Removal of Punctuation: I eliminated most punctuation marks, except for emojis and expressive symbols ('!' and '?'), which are valuable for sentiment analysis.

- Removal of Whitespaces

- Contraction Expanding

- Handling Repeated Letters: Words with repeated letters (e.g., 'coooooool') were processed.

- Removal of 'sent from' Phrases

**Feature Extraction (TF-IDF):**

- For feature extraction, I employed the TF-IDF (Term Frequency-Inverse Document Frequency) vectorization method. This method is highly effective for NLP tasks.

**Feature Selection (Chi-squared):**

- To reduce the number of features and improve efficiency (and overcome overfitting), I implemented chi-squared feature selection. Specifically, I used `SelectKBest(chi2, k=1000)` to select the top 1000 most relevant features.

**Model Training:**

I trained multiple models to compare their performance. My best-performing method was the SVM (Support Vector Machine) classifier, which utilized a Gaussian RBF kernel.

Hyperparameter Tuning: Hyperparameter tuning was crucial for model performance. I focused on two key hyperparameters:

- C (Regularization Parameter): This parameter controls the trade-off between maximizing the margin and minimizing the classification error.

- Gamma (SVM Parameter): Gamma defines the influence range of a single training example.

To find the optimal hyperparameters, I employed GridSearchCV with 5-fold cross-validation. I used balanced accuracy as the scoring metric, which accounts for both positive and negative prediction errors.

**Best Model Parameters:**

- The hyperparameters that yielded the best model performance were C = 100 and gamma = 0.032.

**Model Evaluation Metrics**:

I assessed my model's performance using several metrics, including:

- Accuracy: 0.9412

- Precision: 0.9286

- Recall: 0.9630

- F1-Score: 0.9455

**Conclusion**:

My solution effectively handled data imbalance, implemented robust text preprocessing, and fine-tuned a powerful SVM classifier, resulting in accurate sentiment analysis.