Pouya Mirzaei Zadeh

Emotion Detection Model

Introduction: The project aims to construct a sophisticated machine-learning model that can accurately categorize text data into specific emotions, enhancing understanding of emotional context in written communication.

Data Preprocessing: The initial dataset comprised various text entries associated with an emotion label. The preprocessing phase involved cleaning the text data to ensure uniformity and improve model performance. This process included:

- Removing any HTML tags that may have been present in the text. • Eliminating punctuation and special characters to focus on textual content. • Converting all text to lowercase to maintain consistency and prevent duplication of the same words in different cases.
- Stripping extra spaces to standardize the text format.

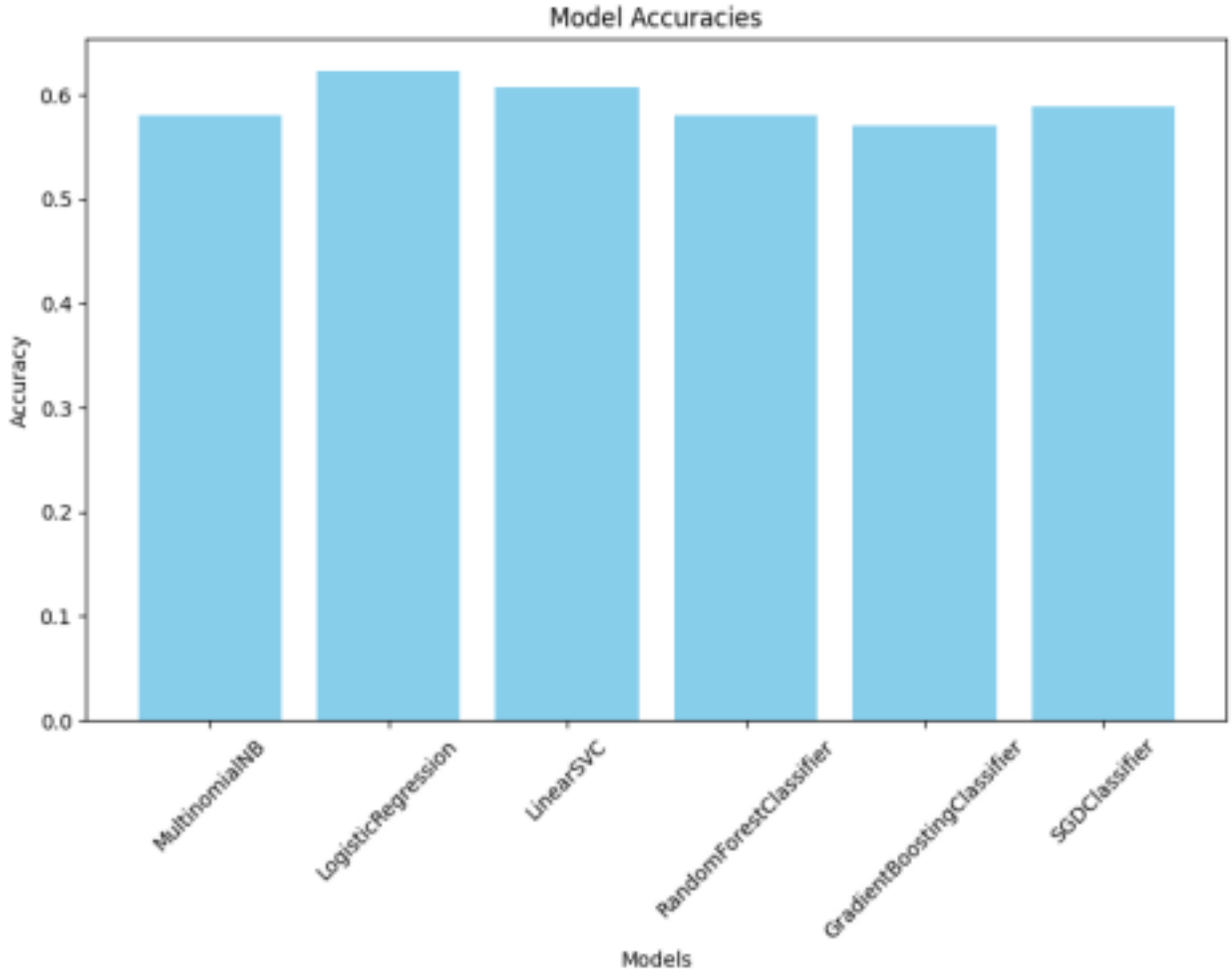
Feature Engineering: Feature engineering is a critical step in preparing data for machine learning. In this project, we employed Term Frequency-Inverse Document Frequency (TF-IDF) vectorization to convert the preprocessed text into a numerical format that machine learning algorithms can interpret. TF-IDF measures the importance of a term within a document relative to its frequency across all documents, thus highlighting words that are more descriptive of the content. The TF-IDF vectorizer was fine-tuned with parameters such as max_features, ngram_range, and use_idf to optimize its performance.

Model Training and Selection: We selected RandomForestClassifier as our base model due to its robustness and ability to handle non-linear relationships. To refine our model, we utilized grid search—a systematic approach to tuning hyperparameters. Grid search iteratively trains the model with different combinations of hyperparameters specified in a predefined grid. For our RandomForestClassifier, we varied parameters like n_estimators (number of trees), max_depth (maximum depth of each tree), and min_samples_split (minimum number of samples required to split an internal node). The grid search process identified the optimal combination as n_estimators=200, max_depth=None, and min_samples_split=5, which provided the best balance between bias and variance.

Model Evaluation: To evaluate our model's performance, we used classification_report from sklearn. metrics, which provided a detailed breakdown of precision, recall, f1-score, and support for each class label. Additionally, we employed cross-validation a technique that divides the dataset into k subsets (folds) and iterates the training and validation process k times. Each fold serves as a validation set once while the remaining

folds form the training set. This method ensures that every data point is used for both training and validation exactly once, offering a comprehensive assessment of the model's performance. Our cross-validation with five folds yielded an average accuracy score of 0.575.

Comparative Analysis: We compared our optimized RandomForestClassifier with other models like MultinomialNB, LogisticRegression, LinearSVC, GradientBoostingClassifier, and SGDClassifier. Each model was evaluated based on its accuracy score to determine which performed best on our dataset.



Test Data Predictions: For predicting emotions on new text data, we applied the same preprocessing steps and transformed the data using our trained TF-IDF vectorizer. The predictions were made using our optimized RandomForestClassifier pipeline.

Conclusion: The project successfully implemented various machine learning models for emotion detection in text data. While all models showed similar accuracy levels, LogisticRegression emerged as the top performer with an accuracy score of 0.6223.