ML HACKATHON

Prepared By:

Kripa S Rai PES1UG22CS291

JEEVAN KUMAR U PES1UG22CS256

K MONIKA PES1UG22CS263

KEERTHANA S PES1UG22CS282

Preprocessing Methods:

1. Label Encoding for Emotion Classification

- **Purpose**: Converts categorical emotion labels into numerical format for model compatibility.
- **Details**: The `LabelEncoder` is applied to the `Emotion` column in `train_df`, assigning each unique emotion a specific integer label. This enables supervised learning algorithms to process emotion classes as numerical values.
- **Class Weights Calculation**: Class weights are calculated using `compute_class_weight` to handle class imbalances, with a dictionary storing weights per class for later model training.

2. Feature Engineering and Fusion for Multimodal Data

- **Feature Concatenation**: The text, audio, and visual features are concatenated into a single array for each sample, creating an `early_fusion_features` column in `train_df`.
- **Resampling**: An `RandomOverSampler` is applied to the concatenated features to balance the dataset, duplicating underrepresented samples in `early_fusion_features`.
- **Train-Test Split**: The resampled dataset is split into training and validation sets, specifically for early fusion models.

3. Scaling and Model Training for Early Fusion Model

- **Standard Scaling**: A `StandardScaler` is used in the model pipeline to normalize the concatenated features. This improves the model's performance by ensuring the feature distribution is standardized.
- **Random Forest Classifier**: A `RandomForestClassifier` is used, with class weights applied from the earlier calculations to address class imbalance.

4. Data Preprocessing for Late Fusion Models

- **Separate Feature Sets**: Each modality (text, audio, visual) is processed independently.
- **Resampling by Modality**: The `RandomOverSampler` is applied separately to text, audio, and visual features, ensuring balanced data across modalities.
- **Train-Test Split**: Each resampled dataset (text, audio, visual) is split into training and validation sets for individual model training.

5. Scaling and Model Training for Late Fusion Models

- **Standard Scaling**: Each modality-specific model pipeline includes a `StandardScaler` to normalize features for that modality.
- **Separate Models**:
  - **Text Model**: Uses a Support Vector Classifier (SVC) with class weights for handling imbalances.
  - **Audio Model**: Uses a `RandomForestClassifier` with class weights.
  - **Visual Model**: Uses another `SVC`, also with class weights.
- **Fusion through Majority Voting**: After model predictions for each modality, a majority voting mechanism is used to determine the final prediction, combining outputs from text, audio, and visual classifiers.

6.Feature extraction:

- **Text features:** are extracted using bert
- **Audio features:** using MFCC
- **Visual features:** extraction with augmentation using OpenCV

7. Submission Preparation and Final Output Handling

- **Label Decoding**: Predicted integer labels are converted back into emotion labels using `label_encoder.inverse_transform`.
- **Submission File Creation**: The results are saved in a CSV format for submission, including error handling for unseen or unexpected labels.

MODELS USED:

1. Early Fusion Model

- **Model Used**: Random Forest Classifier
- **Pipeline Structure**: The early fusion model pipeline includes two primary components: a `StandardScaler` and a `RandomForestClassifier`.
  - **StandardScaler**: Normalizes the concatenated text, audio, and visual features to have zero mean and unit variance, which is important for stabilizing model performance, particularly when feature ranges differ.
  - **RandomForestClassifier**: A tree-based ensemble method that aggregates multiple decision trees to make final predictions. In this setup, it's configured

with class weights to account for imbalanced data, as determined from the `compute_class_weight` function earlier in preprocessing. This model can handle high-dimensional feature spaces well and benefits from feature bagging to reduce overfitting.

- **Training Process**: The early fusion model is trained on the combined and resampled feature space (`early_fusion_features`), allowing the model to learn from text, audio, and visual information simultaneously.
- **Evaluation**: Predictions are generated for the validation set, and the model's performance is evaluated through a classification report to capture metrics like precision, recall, and F1-score.

2. Late Fusion Models

- **Separate Models by Modality**: For late fusion, each modality (text, audio, and visual) is modeled individually. These models are as follows:
  - **Text Model**: Uses a Support Vector Classifier (SVC) with a linear decision boundary to classify text-based features. An SVC is suitable here as it creates a margin of separation between classes and performs well on smaller, structured datasets with balanced scaling. It's configured with class weights, assisting in managing any class imbalance in the text data.
  - **Audio Model**: Uses a `RandomForestClassifier` configured similarly to the early fusion model, with class weights. This model is robust to variations and noise in the audio features and performs feature bagging, which reduces overfitting.
  - **Visual Model**: Another SVC, but for visual features, where the linear boundaries help separate visual data classifications effectively when combined with `StandardScaler` normalization.
- **Training Process**: Each model is trained separately on its respective resampled and normalized feature set (text, audio, or visual). This allows each classifier to specialize in handling its unique modality data.
- **Fusion Technique**: After generating predictions from each model, the late fusion process combines. For each sample, the predictions from all three models are compared, and the class label with the highest count is selected as the final prediction. Majority voting improves robustness by leveraging the strengths of each modality-specific model.

3. Final Predictions and Output

- **Label Decoding**: The final predictions are decoded from integer labels back to categorical emotion labels for interpretability.
- **Output**: The results are formatted into a CSV for final submission.

OUTPUT SNIPPETS:

FINAL INFERENCES:

- The model achieved a high accuracy by combining features from text, audio, and visual data.
- According to the classification report:
  - Each emotion class, including "anger," "joy," "neutral," "sadness," and "surprise," showed high precision, recall, and F1-scores, all around 0.90 or above.
  - The **weighted average F1-score of 0.96** indicates balanced performance across all classes, meaning the model accurately distinguishes between various emotions with minimal misclassifications.