MELD Dataset Preprocessing for Multimodal Emotion Detection

This document explains how to preprocess the MELD dataset (in CSV format) for a multimodal emotion detection project. It includes tasks such as loading the dataset, cleaning the data, selecting relevant features, encoding labels, and performing stratified train/validation/test splits.

1. MELD Dataset Overview

MELD (Multimodal EmotionLines Dataset) is based on dialogues from the TV series 'Friends'. Each utterance in the dataset contains:

- Text (Utterance)
- Emotion (target label)
- Speaker (person speaking)
- Conversation_ID (conversation grouping)
- Utterance_ID (order within the conversation)

2. data_loader.py

This script performs the following tasks:

- Load MELD CSV files using pandas.

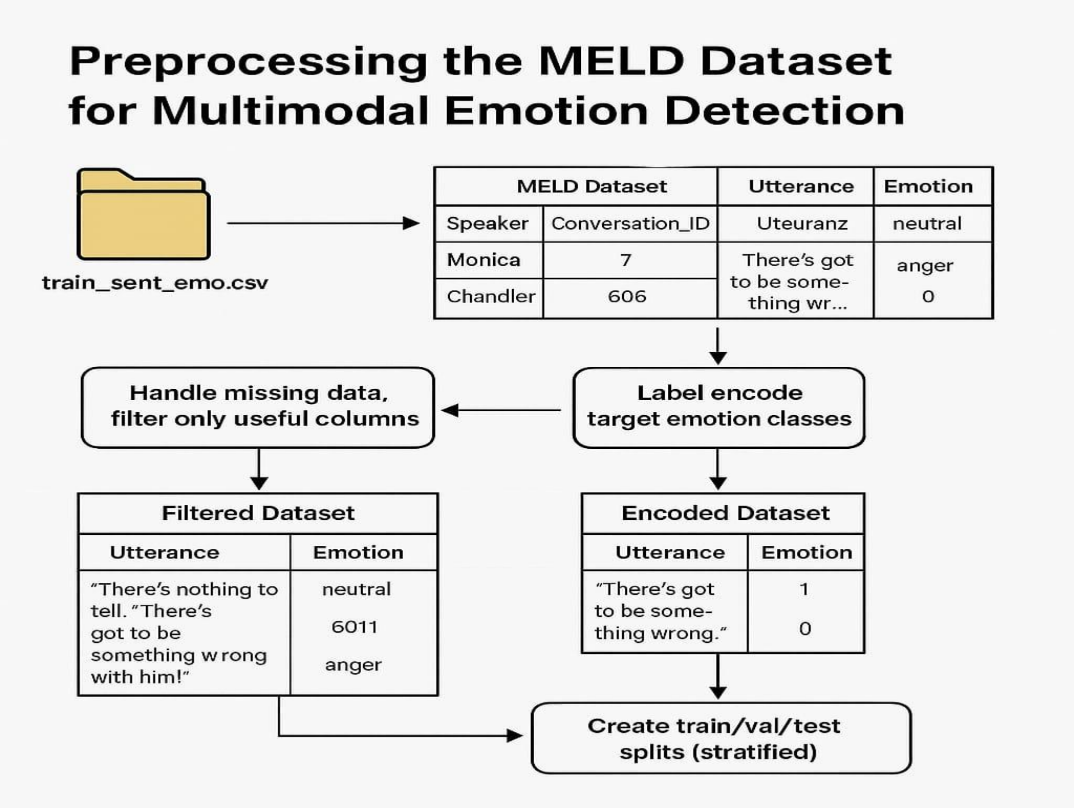
- Filter useful columns: Utterance, Emotion, Speaker, Conversation_ID, Utterance_ID.

- Handle missing data by removing rows with null values.

- Label encode emotion classes using scikit-learn's LabelEncoder.

3. data_splitter.py

This script creates stratified splits of the dataset to ensure each emotion class is equally represented.


Preprocessing the MELD Dataset
for Multimodal Emotion Detection
train_sent_emo.csv
Handle missing data, filter only useful columns
Label encode target emotion classes
Create train/val/test splits (stratified)

MELD Dataset		Utterance	Emotion
Speaker	Conversation_ID	Uteuranz	neutral
Monica	7	There's got to be some-thing wr...	anger
Chandler	606		0

Filtered Dataset	
Utterance	Emotion
"There's nothing to tell. "There's got to be something w rong with him!"	neutral 6011 anger

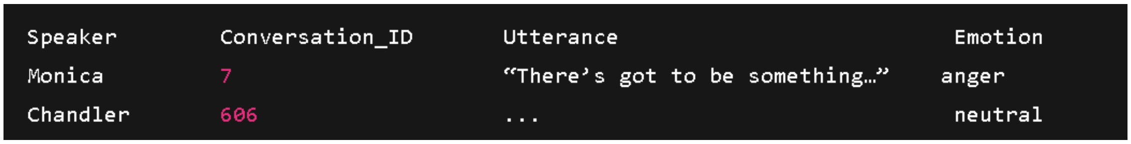
Encoded Dataset	
Utterance	Emotion
"There's got to be some-thing wrong."	1 0

📁 Input: train_sent_emo.csv

This is the **raw dataset** file. It contains:

- Speaker name (e.g., Monica, Chandler)
- Conversation ID (to group utterances)
- Utterance (what the character says)
- Emotion label (e.g., neutral, anger)

👀 Example:



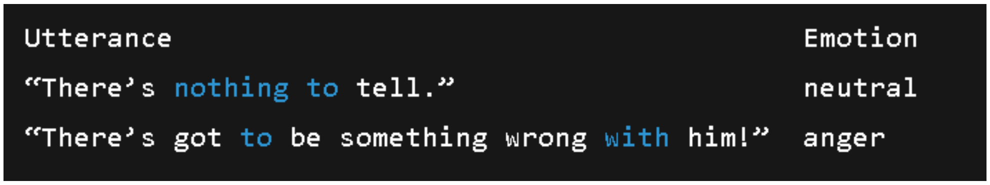
```
Speaker        Conversation_ID      Utterance                           Emotion
Monica         7                    "There's got to be something…"    anger
Chandler       606                  ...                                 neutral
```

🛠️ Step 1: Handle missing data, filter only useful columns

In this step:

- Rows with missing values (NaN) are removed.
- Only the **important columns** are kept:
    - Utterance
    - Emotion
    - (Optionally: Speaker, Conversation_ID, Utterance_ID)

👀 Filtered table example:



```
Utterance                                          Emotion
"There's nothing to tell."                         neutral
"There's got to be something wrong with him!"      anger
```

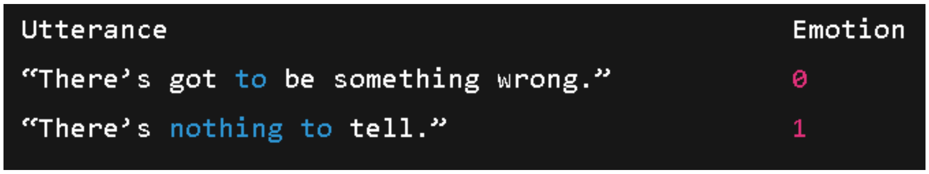
🔢 Step 2: Label encode target emotion classes

Here, the text-based emotion labels like:

- neutral → 1
- anger → 0

are converted into **numerical values**, which machine learning models need.

👀 Encoded table:



```
Utterance                                   Emotion
"There's got to be something wrong."        0
"There's nothing to tell."                  1
```

This uses LabelEncoder from sklearn.

🧪 Step 3: Create train/val/test splits (stratified)

Now that the data is clean and numeric:

- We split it into **training**, **validation**, and **test** sets.
- **Stratified split** means the proportion of each emotion class (e.g., anger, joy, sadness) is kept similar in all three sets.

This is important so that your model doesn't learn from an imbalanced distribution.

4. Summary

Steps involved in processing the MELD dataset:

- Load CSV files.

- Filter only relevant columns.

- Drop rows with missing data.

- Encode emotion labels into numeric format.

- Split the data using stratified sampling into train, validation, and test sets.