Data Pipeline for Speech Emotion Detection

Group 3 - Team Members:

- Team Member 1: Debanjan Saha
- Team Member 2: Ajay Bana
- Team Member 3: Akhil Krishna Nair
- Team Member 4: Sai Venkat Madamanchi
- Team Member 5: Siddharth Banyal
- Team Member 6: Venkatesh Gopinath Bogem

Data pipeline

Table of contents

1. Dataset Information

Our dataset comprises a diverse set of audio samples sourced from reputable databases like RAVDESS, CREMA, TESS, and SAVEE known for their annotated emotional speech and song recordings. To enhance model robustness, custom audio data has been added, spanning various demographics and contexts. This combined approach ensures the model is well-equipped to understand and generate emotional content across a wide range of real-world scenarios.

2. Data Card

Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) Speech audio-only files

This part of the RAVDESS has 1,440 files, with 60 trials for each of the 24 actors (12 female, 12 male). The RAVDESS features professional actors delivering two matching statements in a neutral North American accent. The speech emotions include calm, happy, sad, angry, fearful, surprise, and disgust, each expressed at two intensity levels (normal, strong), along with a neutral expression.

Crowd Sourced Emotional Multimodal Actors Dataset (CREMA-D) Dataset

CREMA-D is a dataset with 7,442 clips featuring 91 actors, including 48 males and 43 females aged 20 to 74, representing diverse races and ethnicities (African American, Asian, Caucasian, Hispanic, and Unspecified). The actors spoke 12 sentences expressing six emotions (Anger, Disgust, Fear, Happy, Neutral, and Sad) at four emotion levels (Low, Medium, High, and Unspecified).

Toronto emotional speech set (TESS) Dataset

Two actresses, aged 26 and 64, spoke a set of 200 target words in the phrase "Say the word _," expressing seven emotions (anger, disgust, fear, happiness, pleasant surprise, sadness, and neutral). Recordings were made, resulting in 2,800 audio files. The dataset is organized with separate folders for each actress, containing their emotions. Each folder includes audio files for all 200 target words in WAV format.

Surrey Audio-Visual Expressed Emotion (SAVEE) Dataset

The SAVEE database features recordings from four native English male speakers (DC, JE, JK, KL), all postgraduate students and researchers at the University of Surrey, aged 27 to 31. Emotions, categorized as anger, disgust, fear, happiness, sadness, and surprise, align with psychological descriptions. We added a neutral category, resulting in recordings of seven emotions. Each emotion is represented by 15 TIMIT sentences, including three common, two emotion-specific, and 10 generic sentences. These sentences are phonetically balanced and vary for each emotion, totaling 30 neutral sentences recorded as neutral.

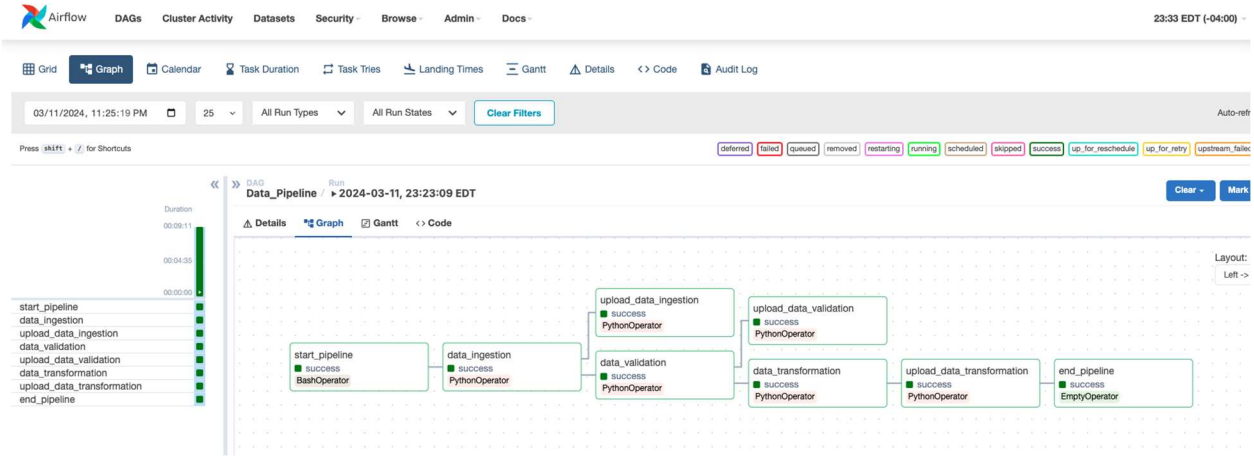
3. Data Sources

The data is taken from RAVDESS, CREMA, TESS, and SAVEE

Organized our data pipeline into modular components, spanning from data ingestion to preprocessing, ensuring our data is model-ready. To guarantee the functionality of each module, we implement Test Driven Development (TDD), enforcing tests for every part of the pipeline.

4. Airflow Setup

Apache Airflow, a Python-powered platform, necessitates installation through pip, database initialization, and configuration file modifications. Following these steps, you launch the scheduler and web server to oversee Directed Acyclic Graphs (DAGs) using a browser-based interface. Within this UI, you articulate tasks and their dependencies to orchestrate workflow automation. The attached Airflow script encapsulates these functionalities.



5. Data Pipeline Components: -

In this project, our data pipeline comprises interconnected modules, each assigned specific tasks for data processing. We leverage Airflow and Docker for orchestration and containerization, with each module acting as a task within the primary data pipeline directed acyclic graph (DAG).

5.1 Downloading Data

In the initial stage, the dataset is downloaded and extracted into the data directory using specific modules. This process involves retrieving data from four distinct sources.

5.1.1 Data Loading

- We use an audio library (librosa) to load the audio files into memory.
- Then verify that the audio data is correctly loaded and accessible for further processing.

5.2 Data Preprocessing

The preprocessing phase involves integrating data from varied sources like RAVDESS and TESS, augmenting data for diversity, and extracting essential features from audio files. This ensures a comprehensive dataset, readying it for effective training by capturing a broad range of emotional expressions and enhancing the model's adaptability to real-world scenarios.

5.2.1 Data Integration

The script efficiently processes audio data from RAVDESS, CREMA, TESS, and SAVEE datasets. It iterates through each dataset, extracting emotion labels and file paths to create individual Data Frames (ravdess_df, Crema_df, Tess_df, Savee_df). These are then integrated into a unified dataset (data_path) by concatenation, capturing a diverse range of emotional expressions. The consolidated dataset is saved as a data frame providing a comprehensive resource for subsequent analysis.
pipeline/airflow/dags/stage_01_data_ingestion.py

5.2.2 Data Augmentation

- Noise:
  - *Purpose:* Introduces random noise to simulate environmental variations.
  - *Effect:* Enhances the model's robustness to real-world conditions.
- Stretch:
  - *Purpose:* Alters the tempo by stretching or compressing the audio signal.
  - *Effect:* Mimics variations in speaking rate, aiding the model in generalizing to different speech speeds.
- Shift:
  - *Purpose:* Temporally shifts the audio signal.
  - *Effect:* Simulates changes in speaker pacing, contributing to the model's adaptability.
- Pitch:
  - *Purpose:* Modifies the pitch of the audio signal.
  - *Effect:* Enables the model to generalize across diverse pitch variations in human speech.

These audio data augmentation techniques, exemplified in the provided code, play a crucial role in diversifying the training dataset and improving the model's performance on tasks like speech emotion detection.

5.2.3 Feature Extraction

- The code conducts audio feature extraction and augmentation for a group of audio files, ultimately saving both the features and their corresponding emotions into a CSV file. Key feature extraction functions include 'zcr' for zero-crossing rate, 'rmse' for root mean square error, and 'mfcc' for Mel-frequency cepstral coefficients, encapsulating essential characteristics of the audio content.
- A central 'extract_features' function harmonizes these diverse features into a unified array. Additionally, the 'get_features' function loads an audio file, performs feature extraction, and applies noise, pitch, and combined augmentations. pipeline/airflow/dags/data_pipeline.py

5.2.4 Data Normalization

- We normalize the extracted features to ensure that they have consistent scales and distributions.
- Our normalization techniques include z-score normalization and min-max scaling.

5.3 Data Splitting

- We finally split the preprocessed data into training, validation, and test sets.
- We also ensure that each set contains a representative distribution of emotions to prevent bias.

5.4 Data Storage

- We save the preprocessed data and corresponding labels in a suitable format (e.g., CSV, parquet) along with compression (e.g, gzip) for easy access during model training.
- We organize the data into directories or files based on the chosen storage format and directory structure.
- Finally, the last stage of our data pipeline uploads the data into Google Cloud Storage (GCS) for later retrieval and use.