Assignment: Developing an Ensemble Model for Detecting Infant Cries, Screams, and Normal Utterances

Objective: The goal of this assignment is to develop a robust audio classification system capable of distinguishing between infant cries, screams, and normal utterances. This will involve training individual models using YAMNet and Wav2Vec2 architectures, creating an ensemble of these models, and deploying the solution within a Temporal workflow.

Tasks: Select an appropriate number of samples in the experimental and control datasets and justify your reasoning. Experimental data includes samples of cry, and scream, whereas control samples include non-cry, non-scream, speech, music, and other indoor environment sounds.

1. **Data Acquisition and Preprocessing:**
    - **Dataset Selection:**
        - **Infant Cry Datasets:**
            - **Infant Cry and Snoring Detection (ICSD) Dataset:** A comprehensive dataset designed for infant cry and snoring detection tasks.
              arXiv
            - **Infant Cry Audio Corpus:** A collection of infant cry sounds built through the Donate-a-Cry campaign.
              Kaggle
            - **Infant's Cry Sound Dataset:** Contains recordings classified into categories like hungry, tired, and uncomfortable.
              Mendeley Data
            - **Baby cry datasets:** A collection of high-pitched human vocalizations indicating various cry emotions.
              AudioSet 1 and AudioSet 2
        - **Screaming Datasets:**
            - **Human Screaming Detection Dataset:** Focuses on the identification and analysis of human screams.
              Kaggle
            - **Screaming Audio Clips from AudioSet:** A collection of high-pitched human vocalizations indicating various emotions.
              AudioSet 3
        - **Normal Utterances Datasets:**
            - **Common Voice Dataset:** A multilingual dataset providing a wide range of normal speech recordings.
            - **LibriSpeech Dataset:** Approximately 1000 hours of 16kHz read English speech derived from audiobooks.
            - **Children's speech:** Collection of a variety of children's single and group speech samples AudioSet 4

- **Data Preparation:** Download and preprocess the audio files to ensure consistency in format (e.g., sampling rate, bit depth). Segment the audio into appropriate lengths and label each segment as 'crying,' 'screaming,' or 'normal.' You could include and use any more data from open-source datasets.

2. **Model Training:**
   - **YAMNet Model:**
     - Fine-tune the pre-trained YAMNet model on the prepared dataset to classify the audio segments into the three categories.
     - Implement necessary modifications to adapt YAMNet for this specific task.
   - **Wav2Vec2 Model:**
     - Fine-tune the pre-trained Wav2Vec2 model on the same dataset for the same classification task.
     - Adjust the model architecture as needed to suit the classification requirements.
3. **Ensemble Model Development:**
   - Combine the predictions of both YAMNet and Wav2Vec2 models to create an ensemble model.
   - Explore various ensemble techniques such as averaging probabilities, majority voting, or training a meta-classifier on the outputs of the individual models.
4. **Training, Testing, and Validation Approach:**
   - **Training Approach:**
     - Split the dataset into training, validation, and test sets (e.g., 70% training, 15% validation, 15% testing).
     - Apply data augmentation techniques to enhance model robustness.
     - Use cross-validation to ensure the model's generalizability.
   - **Testing Approach:**
     - Evaluate each model on the test set to assess performance metrics such as accuracy, precision, recall, and F1-score.
     - Compare the performance of individual models and the ensemble model.
   - **Validation Approach:**
     - Monitor model performance on the validation set during training to prevent overfitting.
     - Use early stopping based on validation loss or accuracy improvements.
5. **Justification***:* This structured approach ensures that the models are trained on a substantial portion of the data, validated to tune hyperparameters and prevent overfitting, and tested on unseen data to evaluate real-world performance.
6. **Loss Function Justification:**
   - Choose appropriate loss function , justifying its adoption based on the data samples you are using and also on experimental and control split.
7. **Performance Metrics:**
   - Generate a report detailing the performance of each model and the ensemble.
   - Include metrics such as confusion matrices, ROC curves, and classification reports.
   - Analyze the results to identify strengths and areas for improvement.

8. **Deployment with Temporal:**
    - Design a workflow using Temporal to handle the audio classification process.
    - Implement tasks within the workflow for:
        - Receiving and preprocessing audio input.
        - Running the ensemble model to classify the audio.
        - Storing and managing the results.
    - Ensure the workflow is can handle real-time audio streams if necessary of at least 5 seconds duration upto 15 seconds each chunk.

Deliverables:

- **Code Repository:** A well-structured repository containing all code, including:
    - Data preprocessing scripts.
    - Model training and evaluation scripts for both YAMNet and Wav2Vec2 models including training and validation graphs.
    - Ensemble model implementation.
    - Temporal workflow code.
- **README File:** A comprehensive guide detailing:
    - Project overview and objectives.
    - Setup instructions, including environment setup and dependencies.
    - Inference code instructions to run end-to-end given a test audio file.

- Share the code via GIT and mail to Manu Kohli manu@frontera.health