Troy Sabia

C964

Task C - Design and Development

***1. one*** *descriptive method and* ***one*** *non-descriptive (predictive or prescriptive) method*

We will use descriptive analysis by plotting the MFCC feature distributions to better understand the dataset before training. The convolutional neural network itself is the prescriptive method. By training it on the metal and non\_metal datasets we will be able to predict whether a given music file belongs in the metal genre or not.

*2. collected or available datasets*

While the GTZAN dataset exists, it is only a few hundred samples. We expect to need a much larger dataset and will acquire commercial music files that we can then extract features from and augment the dataset.

*3. decision support functionality*

The question at play here is “is this file suitable for inclusion into the all-metal music catalog?” This tool directly supports that decision by programmatically filtering and categorizing musical tracks, reducing manual reliance and preserving genre integrity.

*4. ability to support featurizing, parsing, cleaning, and wrangling datasets*

**Featurizing:** the preprocess\_dataset.py file will extract MFCCs from the music files (first for training, then later for classification) using the torchaudio library.

**Parsing:** reading and converting MP3s to WAV files and conversion to mono is handled by using torchaudio library and the external Windows ffmpeg application.

**Cleaning:** resampling, padding, truncation all handled by torchaudio.

**Wrangling:** For training purposes music files were manually organized into separate datasets (training, validation, and testing folders) and the save\_mfcc\_cache\_with\_padding.py file processes these and programmatically populates the /mfcc\_cache folder. The process\_directory.py file handles MFCC extraction and directory creation/navigation and file transfer during sorting.

*5. methods and algorithms supporting data exploration and preparation*

This project employs methods for data exploration and preparation such as class distribution analysis, MFCC feature inspection, and quality checks for corrupted or mislabeled files. Algorithms for resampling, padding, and feature extraction create a consistent database suitable for training.

*6. data visualization functionalities for data exploration and inspection*

Waveforms, spectrograms, and MFCC heatmaps will be generated to illustrate feature extraction. Training and validation loss curves will be plotted to monitor and evaluate model performance, and confusion matrices and ROC curves will help analyze classification effectiveness.

*7. implementation of interactive queries*

Creating a fully interactive application (in the sense of natural language support or adjusting thresholds) is beyond the scope of this proposal, but a commandline inteface and instruction set that allows users to place their music files in a given directory and have the application classify the files in that given directory, is implemented.

*8. implementation of machine-learning methods and algorithms*

Machine learning is implemented through a convolutional neural network trained on MFCC features extracted from audio clips. The model uses BCEWithLogitsLoss with class weighting to handle imbalanced data, the Adam optimizer with learning-rate scheduling for efficient convergence, and dropout for regularization. These algorithms together are the predictive engine of this product.

*9. functionalities to evaluate the accuracy of the data product*

The product will include evaluation functionalities such as accuracy, precision, recall, F1 score, ROC-AUC, and confusion matrices. These metrics ensure that classification performance is rigorously validated against a held-out test set, enabling both quantitative accuracy assessment and qualitative inspection of error types

*10. industry-appropriate security features*

Security is supported by having only non-reversible MFCC feature tensors, preventing redistribution of copyrighted material. Data directories and trained model files will be protected with access controls, and future deployment will leverage containerization for isolation and secure execution.

*11. tools to monitor and maintain the product*

Monitoring and maintenance is supported by training logs and validation accuracy training. The system allows for periodic retraining when new datasets are provided to ensure that the model remains accurate as the music catalog expands and potential increase of functionality.

*12. a user-friendly, functional dashboard that includes******three******visualization types*

A simple dashboard will be developed using a lightweight tool such as Streamlit or Plotly. The dashboard will show three visualizations - a class distribution bar chart to show dataset composition, training and validation loss curves to illustrate model learning, and a confusion matrix to visualize prediction performance and accuracy. These visualizations make the operation transparent and user-friendly.