Project Report

Moodify

Speech Emotion - Based Music Recommendation System

TABLE OF CONTENTS

1. INTRODUCTION

In today's digital era, music recommendation systems play a key role in delivering personalized listening experiences. However, traditional systems often rely solely on factors like genre or popularity, overlooking the user's emotional state. This project addresses that gap by introducing a speech-based emotion-aware music recommendation system. By analyzing emotional cues in a user's voice, the system accurately detects their current mood and recommends music that aligns with their emotional state. Leveraging speech emotion recognition techniques and seamless integration with the Spotify platform, the system delivers dynamic and mood-driven music suggestions. This innovative approach transforms how users interact with music by offering a more emotionally intelligent and immersive listening experience through their voice alone.

2. PROJECT OBJECTIVE

The goal of this project is to develop an AI-powered music recommendation system that detects human emotions from speech and suggests mood-appropriate music in real time. The system integrates multiple intelligent subsystems, including audio preprocessing, emotion detection through a deep learning-based Speech Emotion Recognition (SER) model and content-based music recommendation using the Spotify API via the Spotipy Python client.

The core concept is to capture a user's emotional state from their voice input and map it to a corresponding music genre or mood, thereby enhancing their listening experience through emotionally aware recommendations. This is accomplished by extracting relevant audio features (such as MFCCs, Chroma, and Mel Spectrogram) from the speech input, processing them through a Convolutional Neural Network (CNN) model trained for emotion classification, and using the predicted emotion to guide music selection.

Rather than building a recommendation engine from scratch, the system leverages Spotify's powerful music platform to fetch real-time track suggestions using content-based filtering. This approach allows the system to recommend songs based on audio features and mood alignment, ensuring quick integration and a wide range of music choices.

The proposed system aims to demonstrate how effective computing and machine learning can be combined with commercial APIs to create a more engaging, personalized, and emotionally intelligent digital music experience.

3. REQUIREMENTS

Python 3.11 – Core programming language used to develop the entire system.
VS Code / Google Colab – Development environments used for writing and running code.
librosa - Used for audio processing and feature extraction from speech signals.
numpy - Provides support for numerical operations and handling arrays.
pandas - Used for data manipulation and managing datasets.
matplotlib, seaborn - Libraries for visualizing data, model performance, and analysis results.
scikit-learn - Used for preprocessing, model evaluation, and machine learning utilities.
imblearn - Provides the SMOTE technique to handle class imbalance.

tensorflow - Framework for building and training the CNN emotion classification model.
spotipy - Python client for accessing the Spotify Web API to fetch music recommendations.
streamlit - Lightweight UI framework to build an interactive web app for the project.

4. KEY FEATURES:

End-to-End Audio Processing Using Librosa
Audio signals are pre-processed and analyzed using the Librosa library, which extracts meaningful features such as MFCCs, Chroma, Mel Spectrogram, and more for emotion classification.

Emotion Prediction Using a CNN Model
A custom-designed Convolutional Neural Network (CNN) processes the extracted features to predict the emotional state (Happy, Sad, Angry, Neutral, Fear, Disgust) from speech input with high accuracy.

5-Fold Cross-Validation for Model Evaluation
To ensure robustness and generalization, the model was validated using 5-fold cross-validation, providing a reliable estimate of prediction performance across different data splits.

SMOTE for Handling Class Imbalance
Synthetic Minority Over-sampling Technique (SMOTE) was applied to balance the dataset by generating synthetic examples for underrepresented emotion classes, improving model fairness and accuracy.

Spotify Integration for Music Recommendations (Content-Based Filtering)
Based on the predicted emotion, the system connects to Spotify via the Spotipy API and recommends mood-matching music using a content-based filtering approach, mapping emotions to genres.

Clean Visualizations for Model Insights
Model performance is presented using intuitive visualizations, including training/validation accuracy and loss plots, as well as a confusion matrix to highlight classification strengths and weaknesses.

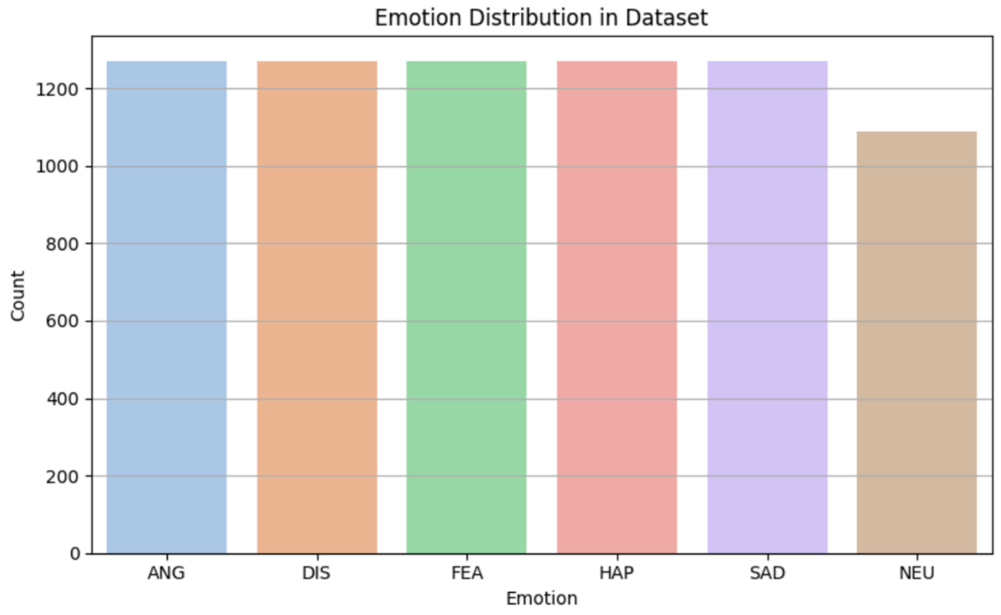
5. DATA PRE-PROCESSING

Dataset Used: CREMA-D (Crowd-sourced Emotional Multimodal Actors Dataset)

Source: https://www.kaggle.com/datasets/aliaaebrahim/crema-d-public

Total Audio Clips: The dataset contains 7,442 audio clips featuring acted emotional expressions providing a rich foundation for training deep learning models.
Emotion Classes: It includes six distinct emotions - Neutral, Happy, Sad, Angry, Fear, and Disgust allowing for a wide range of emotional classification tasks.

Emotion Levels: Each emotion is labeled with one of four intensity levels: Low, Medium, High, or Unspecified Participants: 91 professional actors (48 male, 43 female)
Diversity: The dataset includes speakers from various age groups and ethnic backgrounds, which adds to its robustness and generalizability across demographics.
File Naming Convention: Each audio file is named using a structured format:
"ActorID_Statement_Emotion_Intensity.wav"
For example, the file 1001_DFA_ANG_XX.wav indicates an Angry emotion expression from Actor 1001, speaking the DFA sentence with unspecified intensity.



6. MODEL OVERVIEW

The system architecture consists of three major components:

1. Audio Processing Module
2. Speech Emotion Recognition Module
3. Music Recommendation System

6.1. AUDIO PROCESSING MODULE

6.1.1 Load and Prepare Data:

All audio file paths were retrieved from the **AudioWAV** directory, which contained the dataset of recorded speech samples. Each audio file name followed a structured naming convention that encoded important metadata, including the speaker ID, sentence type, emotion label, and intensity level. The emotion labels were extracted by parsing the filenames, allowing the system to associate each audio clip with its corresponding emotional category. To prepare the data for classification, the extracted labels were converted into numerical form using LabelEncoder from the scikit-learn library. This transformation ensured that the categorical emotion labels could be effectively used as targets during model training.

6.1.2 Audio Feature Extraction:

A diverse set of audio features was extracted from .wav files using the **Librosa** library to effectively capture emotional characteristics embedded in speech.

Time-Domain Features:
Zero Crossing Rate and Root Mean Square Energy (RMSE) were used to capture temporal signal properties such as voicing activity and intensity which are essential for detecting emotional arousal in speech.

Spectral Shape Features:
Features like Spectral Centroid, Spectral Bandwidth and Spectral Rolloff were extracted to analyze the distribution and energy spread of frequencies helping to distinguish between bright, sharp tones and mellow, soft ones.

Spectral Contrast and Tonal Features:
Spectral Contrast and Tonnetz were included to capture variations in spectral peaks and harmonic relationships, which are indicative of emotional tone and musicality in vocal expressions.

MFCCs (Mel-Frequency Cepstral Coefficients):
The model utilized both the mean and standard deviation of 40 MFCC coefficients to represent the timbral texture of speech, providing detailed insights into the vocal tract characteristics that vary across emotional states.

Chroma-Based Features:
Chroma Features were computed to analyze pitch class intensities, capturing harmonic and melodic cues that are often subconsciously linked to specific emotions.

Time-Frequency Representations:
The Mel Spectrogram was used to obtain a perceptually relevant time-frequency view of the audio, while harmonic and percussive components were separated to analyze sustained versus transient sound properties, enriching the emotion recognition process.

6.1.3 Data Augmentation Techniques:

To make our model more robust and avoid overfitting, we applied data augmentation techniques like noise injection, time shifting, and pitch shifting. These helped us create varied versions of the same audio clip, which improved the model's ability to generalize on new data.

Class Balancing with SMOTE: Our dataset had an uneven number of samples for different emotions. For example, emotions like Neutral had fewer examples. We used SMOTE—Synthetic Minority Oversampling Technique—to generate new synthetic samples for underrepresented classes, helping the model learn all emotions more fairly and prevent any bias

6.2. SPEECH EMOTION RECOGNITION

A Convolutional Neural Network (CNN) model has been developed to learn and classify emotional cues from speech by analyzing a variety of extracted audio features including Mel-spectrograms, MFCCs (Mel-Frequency Cepstral Coefficients), and Chroma features. By learning intricate patterns in tone, intensity and rhythm within the audio signal, the model can accurately identify emotions such as Happy, Sad, Angry, Neutral, Fear and Disgust. This enables the system to interpret the speaker's emotional state with a high degree of precision.

6.2.1 CNN MODEL ARCHITECTURE:

In this project, two methods were implemented for emotion recognition using a Convolutional Neural Network (CNN) architecture. The CNN model was specifically designed to process audio features extracted from speech, such as MFCCs and Mel Spectrograms, which represent the temporal and spectral properties of the voice signal.

The architecture consisted of two convolutional blocks, each containing a 1D Convolutional layer, followed by Batch Normalization, Max Pooling, and a Dropout layer. The convolutional layers help in capturing local audio patterns such as pitch, tone, and rhythm changes. Batch normalization was used to stabilize and accelerate training by normalizing the output of convolutional layers, while max pooling reduced the dimensionality, preserving dominant features. Dropout layers were added to prevent overfitting by randomly deactivating neurons during training.

The output of the final convolutional block was passed through a Flatten layer to convert the two-dimensional feature maps into a one-dimensional vector. This was followed by fully connected layers consisting of a Dense layer, another Dropout layer, and an Output layer. The Dense layer learned high-level abstract representations from the extracted features. The final output layer used a Softmax activation function to provide the probability distribution across different emotion classes.

The model was compiled using the Adam optimizer, which provides efficient training with adaptive learning rates. The categorical crossentropy loss function was used, as the task involved multi-class emotion classification. Accuracy was used as the primary evaluation metric.

This model effectively captured emotional cues from speech and provided accurate classification results, making it suitable for downstream tasks like emotion-based music recommendation.

Method 1: Comprehensive Feature Extraction and CNN Model

Method 1 employs a detailed feature extraction pipeline that includes Mel-Frequency Cepstral Coefficients (MFCCs), Chroma features, Mel spectrogram, Tonnetz, Zero-Crossing Rate (ZCR), Root Mean Square Error (RMSE), and various spectral features. These features capture a wide range of audio characteristics, from timbral properties to harmonic and rhythmic components. The model is built using a 1D Convolutional Neural Network (CNN), which is complemented with dropout and batch normalization to prevent overfitting and improve generalization. To evaluate performance, the method uses Stratified 5-fold cross-validation, ensuring that each fold maintains the class distribution of the dataset. After training, the best model is selected based on evaluation metrics, such as the classification report and confusion matrix. The model is trained on augmented audio data and a SMOTE-balanced dataset, which helps address any class imbalances and ensures a fair representation of all emotion classes. Finally, after predicting the emotion, it is mapped to a corresponding genre, and relevant music tracks are recommended via the Spotify API.

Method 2: Enhanced Data Augmentation and Optimization

Method 2 builds upon the approach of Method 1, introducing several optimizations to enhance performance. The method incorporates Log-mel spectrogram as the feature extraction technique, which is particularly effective for capturing the frequency characteristics of audio signals. Additional data augmentation strategies, including pitch shifting, noise addition, and time stretching, are used to further diversify the training data and improve the model's robustness to various distortions in real-world scenarios. The model architecture remains similar to that of Method 1, utilizing a 1D CNN, but it also introduces a cosine annealing learning rate scheduler, which adjusts the learning rate during training to facilitate more stable convergence and better performance. Similar to Method 1, SMOTE is used for class balancing, with the added option of applying class weights to the loss function, which helps the model handle class imbalances more effectively. The final predictions are made by selecting the best model after cross-validation, and music track recommendations are generated based on the predicted emotion and genre, utilizing the Spotify API for personalized music suggestions.

6.2.2 MODEL TRAINING CONFIGURATION

To ensure effective training and optimal performance of the speech emotion recognition model, several key hyperparameters and techniques were carefully configured:

The learning rate was set to 0.0003 in the Adam optimizer, a popular adaptive optimization algorithm. This relatively low learning rate was chosen to ensure gradual and stable convergence, allowing the model to fine-tune its weights over time without overshooting minima in the loss function.

The batch size was fixed at 32, which offers a good balance between training speed and memory usage. It allows the model to update weights efficiently while maintaining stable gradient estimates during backpropagation.

Training was conducted over 100 epochs, meaning the model iterated through the entire training dataset 100 times. This provided sufficient opportunities for the model to learn patterns in the data, while still allowing early stopping mechanisms to intervene if needed.

The convolutional layers in the model were equipped with 64 and 128 filters, respectively. These varying filter sizes enabled the model to learn and detect features at different levels of complexity, such as simple tone variations in earlier layers and more abstract speech patterns in deeper layers. A kernel size of (3, 3) was used to capture fine-grained local features in the input data.

To prevent overfitting, a dropout rate of 0.5 was applied to the fully connected (dense) layers. Dropout randomly deactivates a fraction of neurons during training, encouraging the model to learn more robust features by preventing co-adaptation of neurons.

To validate the model's generalizability and performance, K-Fold Cross-Validation was employed with 5 folds. In this method, the dataset was split into five equal parts—four parts were used for training, and one part was reserved for testing. This process was repeated five times, ensuring that each sample in the dataset was used for both training and validation exactly once. This approach reduces the variance in performance evaluation and provides a more robust assessment of the model.

Two essential callbacks were integrated during training:

1. ModelCheckpoint was used to automatically save the model with the best validation accuracy during training, preventing the loss of an optimal model due to overfitting in later epochs.

2. EarlyStopping monitored the validation loss and halted training if there was no improvement for a specified number of consecutive epochs. This helped avoid unnecessary training and reduced the risk of overfitting.

Additionally, a LearningRateScheduler was employed to dynamically adjust the learning rate during training using a cosine annealing schedule. This approach gradually decreases the learning rate in a cosine curve pattern, helping the model converge more smoothly and potentially improving final accuracy by avoiding sharp learning rate drops.

Together, these training strategies ensured that the model was not only accurate but also generalizable and efficient in learning emotional patterns from speech data.

6.2.3 CNN MODEL EVALUATION:

To ensure that the speech emotion recognition model generalizes well across unseen data and is not biased by the training set, **5-fold cross-validation** was employed. This technique divides the dataset into five equal parts (folds) and our model is trained and validated five times each time using a different fold for validation and the remaining four for training.

a)Stratified Splitting

The dataset was split into five folds using stratified sampling, which ensures that the distribution of emotion classes (Happy, Sad, Angry, Neutral, Fear, Disgust) remains consistent in each fold. This is crucial for balanced evaluation, especially in cases where certain emotions may be underrepresented. By preserving class proportions across folds, stratification ensures that the model's performance metrics are reliable and not skewed due to class imbalance.

b)Training and Validation

For each of the five iterations, The model is trained on four folds of the data (80%). It is then validated on the remaining fold (20%). This rotation continues until each fold has been used once as the validation set. This process ensures that every data point is used for both training and validation, promoting a fair evaluation of the model's performance on different subsets of data.

c) Performance Tracking

During each iteration of the cross-validation process, key metrics such as: Training accuracy, Validation accuracy, Training loss, Validation loss are recorded. This helps in monitoring the model's learning behavior and provides insights into its consistency, convergence rate and potential overfitting across different data splits.

d) Best Model Selection

After completing all five folds, the model that achieved the highest validation accuracy is identified and saved as the best-performing model. This model is then used for final testing or deployment. This approach ensures that the final model chosen is not only trained on a majority of the data but also validated effectively, making it more robust and reliable for real-world applications.

6) Plot Accuracy and Loss

Training and validation accuracy and loss are plotted for each fold. Consistent gaps between training and validation metrics across epochs are analyzed.

Final Evaluation on All Data

For the final evaluation, the best-performing model obtained during the cross-validation phase was loaded and used to assess its effectiveness on the entire dataset. The model was employed to predict the emotion classes of all samples, and its outputs were evaluated using standard classification metrics. A comprehensive classification report was generated, detailing precision, recall, F1-score, and support for each emotion class, thereby providing a clear picture of the model's predictive performance. Additionally, a confusion matrix was constructed to visually represent the number of correct and incorrect predictions across different classes. This matrix offered valuable insights into how well the model distinguished between various emotions and where potential misclassifications occurred.

6.3. MUSIC RECOMMENDATION SYSTEM

To enhance the personalization of music recommendations, this project integrates an emotion-driven approach with the Spotify Web API using the Python library Spotify. The goal is to provide users with music suggestions that align with their emotional state, rather than relying solely on traditional metrics like listening history or song popularity.

A key component of the system is the mood mapping strategy, where each detected emotion is linked to a corresponding music genre using a predefined mood_map dictionary. For example, emotions such as Happy are associated with Pop, Sad with Classical, Angry with Rock, Neutral with Chill, Fear with Metal and Disgust with Emo. This mapping ensures that each emotional state is matched to a genre that is likely to resonate with or help regulate that mood.

The system uses content-based filtering as the primary recommendation strategy. Unlike collaborative filtering, which depends on user interactions and preferences, content-based filtering relies on the characteristics of the predicted emotion. After identifying the user's emotion from a speech input, the system maps it to a genre and queries Spotify for relevant tracks. This method leverages music attributes like tempo, mood, and energy levels to ensure the recommendations are emotion-appropriate.

The Spotify API integration is achieved using Spotipy. First, Spotify developer credentials are set up, and the system authenticates using OAuth. Once authenticated, the API is used to search and retrieve tracks from the Spotify library. Upon receiving the predicted emotion, the system connects to Spotify and fetches the top 5 recommended tracks from the corresponding genre using Spotify's search or recommendation endpoints. This provides a seamless and dynamic music listening experience tailored to the user's current mood.

The "Predict and Recommend" module serves as the core operational unit. It accepts an input .wav file, processes the audio to extract relevant features, and uses a pre-trained deep learning model to predict the emotional state of the speaker. The module then prints out the probabilities of each possible emotion and recommends music that corresponds to the most probable emotion.

To validate the system's effectiveness, a testing script is included that runs the full prediction and recommendation pipeline on a set of example audio files. This not only helps assess the model's classification accuracy but also demonstrates how accurately the music suggestions reflect the user's mood based on real audio inputs.

By combining speech-based emotion recognition with Spotify's powerful music catalog, this system offers a unique, emotionally intelligent music recommendation experience. It highlights how AI and emotional computing can work together to deliver a more human-centered digital interaction.

7. RESULTS

A comparative analysis was conducted between two different approaches (Method 1 and Method 2) to assess their effectiveness in emotion recognition from speech.

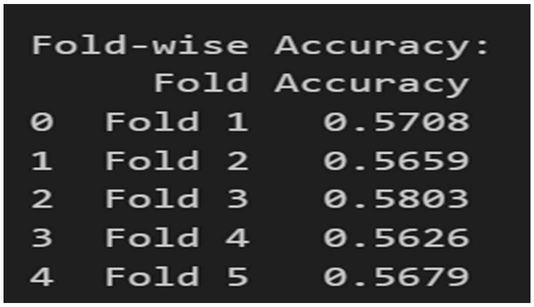
In terms of overall accuracy, Method 2 demonstrated slightly better performance compared to Method 1, particularly in converging and stabilizing training and validation accuracy. The plotted accuracy graphs for both methods show that the curves for training and validation accuracy eventually align without signs of overfitting. Similarly, the loss curves exhibit a rapid decrease during initial epochs and later stabilize, indicating that the model was able to learn efficiently without significant divergence or instability.

Despite the marginally higher accuracy of Method 2, Method 1 outperformed it in other key evaluation metrics. The classification report revealed that Method 1 achieved higher precision, recall, and F1-scores across most emotion classes. Moreover, the confusion matrix for Method 1 showed fewer misclassifications, with stronger true positive rates, especially for Neutral (NEU) and Angry (ANG) classes. However, some confusion remained between Fear (FEA) and Sad (SAD), reflecting the intrinsic similarity in vocal expressions for these emotions.

On the other hand, Method 2 exhibited a slight decline in performance, particularly with increased misclassifications in the SAD and FEA classes. This indicates that while Method 2 generalized well in terms of accuracy, it faced challenges in distinguishing between certain emotionally similar categories, especially on unseen data.

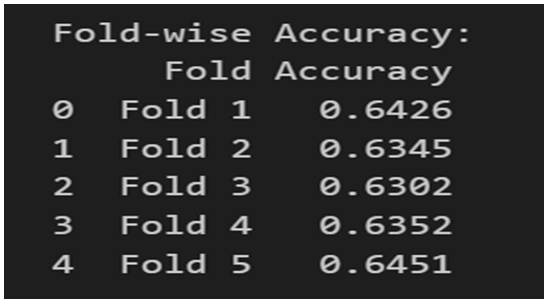
Overall, Method 1 showed stronger and more consistent performance in terms of both classification accuracy and the model's ability to differentiate between closely related emotions.

7.1 Accuracy (5-Fold Cross Validation)



```
Fold-wise Accuracy:
     Fold  Accuracy
0  Fold 1    0.5708
1  Fold 2    0.5659
2  Fold 3    0.5803
3  Fold 4    0.5626
4  Fold 5    0.5679
```

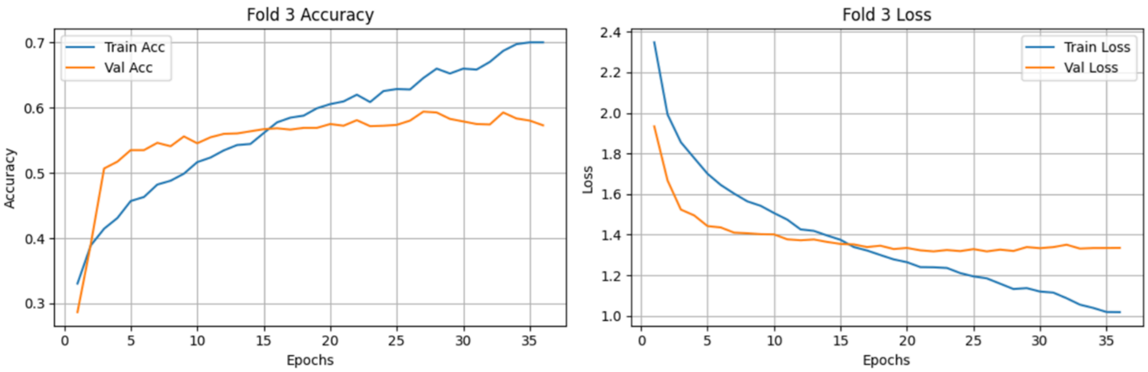
METHOD 1



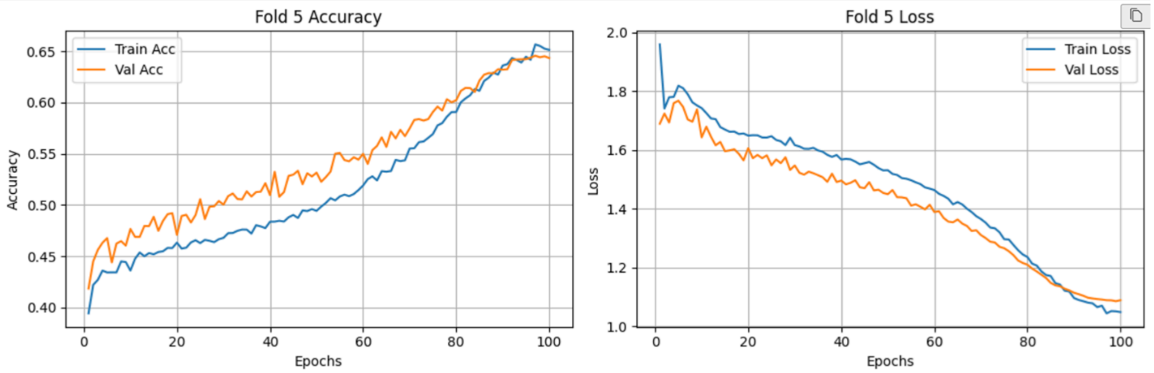
```
Fold-wise Accuracy:
     Fold  Accuracy
0  Fold 1    0.6426
1  Fold 2    0.6345
2  Fold 3    0.6302
3  Fold 4    0.6352
4  Fold 5    0.6451
```

METHOD 2

7.2 Accuracy,Loss Plots

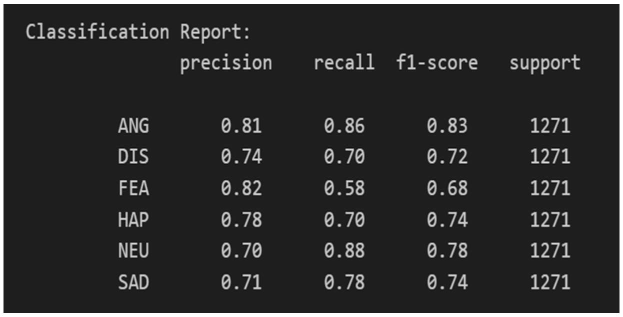


METHOD 1



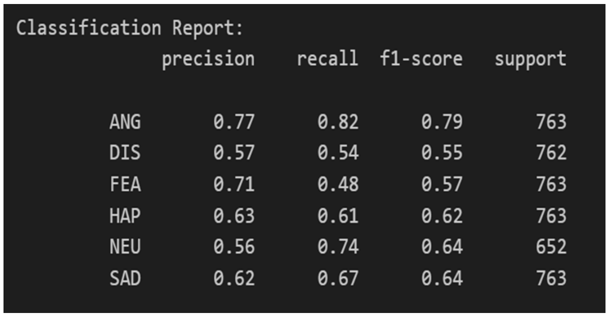
METHOD 2

7.3 Classification Report



```
Classification Report:
              precision    recall  f1-score   support

         ANG       0.81      0.86      0.83      1271
         DIS       0.74      0.70      0.72      1271
         FEA       0.82      0.58      0.68      1271
         HAP       0.78      0.70      0.74      1271
         NEU       0.70      0.88      0.78      1271
         SAD       0.71      0.78      0.74      1271
```

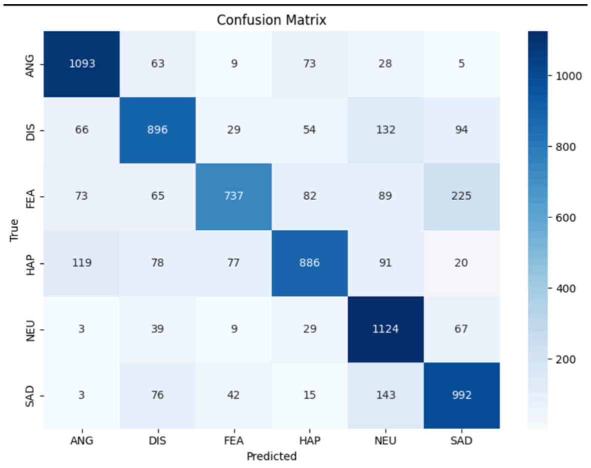


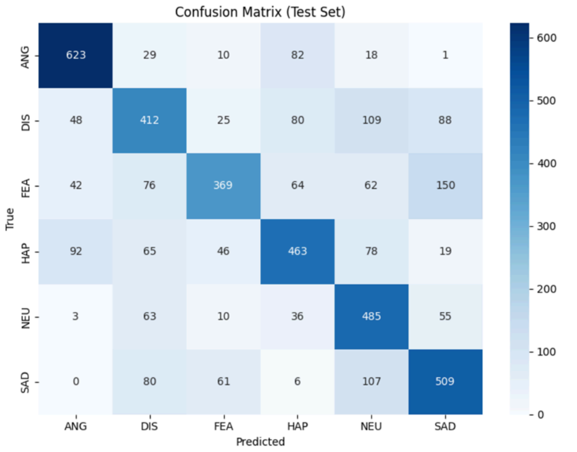
```
Classification Report:
              precision    recall  f1-score   support

         ANG       0.77      0.82      0.79       763
         DIS       0.57      0.54      0.55       762
         FEA       0.71      0.48      0.57       763
         HAP       0.63      0.61      0.62       763
         NEU       0.56      0.74      0.64       652
         SAD       0.62      0.67      0.64       763
```

METHOD 1 **METHOD 2**

7.4 Confusion Matrix

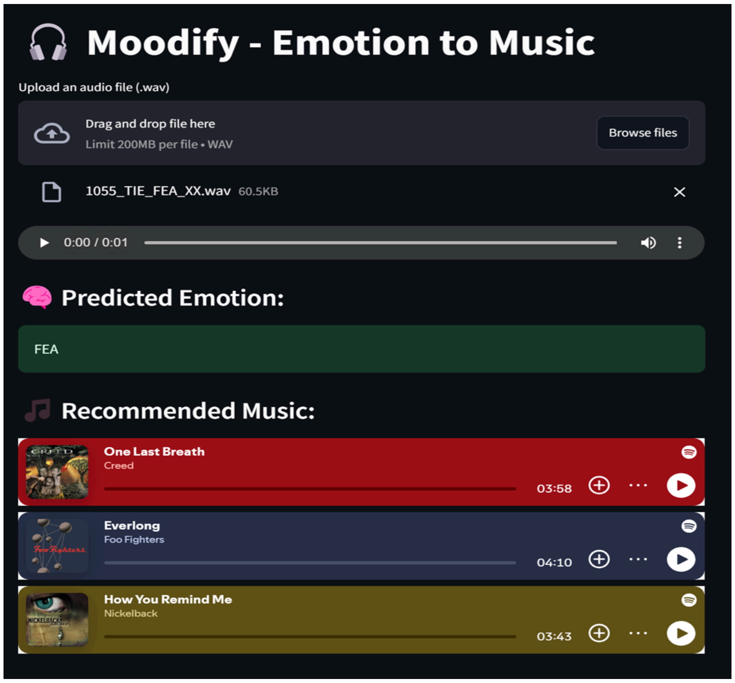




METHOD 1 **METHOD 2**

8. TEST INTERFACE - APPLICATION UI

This is a test implementation of the UI interface based on the project. On the top of the screen there is an option to upload a wav file with emotion intonated speech. In the future, there are plans to make this a live speech emotion recognition field. Once the file is uploaded, the model runs and makes a prediction on the emotion in the speech. Then based on the detected emotion, we mine spotify data to find top tracks for the particular genre using the spotify API. Once the Spotify API returns a list of titles, they are displayed in the recommended music section with an option to play them embedded inside the application.


Moodify - Emotion to Music

Upload an audio file (.wav)

Drag and drop file here
Limit 200MB per file • WAV

1055_TIE_FEA_XX.wav 60.5KB

0:00 / 0:01

Predicted Emotion:

FEA

Recommended Music:

One Last Breath
Creed
03:58

Everlong
Foo Fighters
04:10

How You Remind Me
Nickelback
03:43

9. LIMITATIONS

Despite the successful implementation of the system, several challenges impacted its overall performance. The model's accuracy was limited to approximately 65%, which can be attributed to overlapping emotional characteristics in speech and constraints in feature representation. Class imbalance also posed a significant issue, particularly with underrepresented emotions such as Fear, resulting in skewed model predictions.

Furthermore, the high computational demands of processing audio data and training deep learning models led to extended training durations and increased resource consumption. Another notable limitation was the lack of diversity in the dataset, particularly with respect to speaker accents and variations, which adversely affected the model's ability to generalize across different user profiles.

Finally, the process of audio feature extraction proved to be complex and time-consuming, requiring in-depth research to identify and implement effective techniques that could meaningfully contribute to emotion classification.

10. KEY LEARNINGS

a. This project provided valuable insights into the intersection of speech processing, affective computing, and intelligent system design. A core learning was understanding the correlation between speech and emotion—specifically, how vocal attributes like pitch, tone, and rhythm carry emotional cues that can be effectively captured and interpreted by computational models.
b. In terms of technical development, the project involved extensive audio feature engineering using the Librosa library. Features such as MFCCs, chroma vectors, and spectrograms were successfully extracted and analyzed to differentiate emotional states. The implementation of a 1D Convolutional Neural Network further strengthened understanding of deep learning concepts, including the use of dropout layers, batch normalization, and L2 regularization to enhance model generalization.
c. Addressing class imbalance through SMOTE proved essential in ensuring fair classification across all emotion categories, particularly those underrepresented in the dataset. In parallel, the integration of the Spotify API using Spotipy provided practical experience in combining AI with real-world applications, enabling personalized music recommendations based on detected emotions.
d. Model evaluation techniques such as stratified 5-fold cross-validation, confusion matrices, and accuracy/loss plots were applied to assess and refine performance systematically. Finally, the project demonstrated the real-world potential of affective computing in building emotionally aware systems that can enrich user engagement and interaction.

11. CONCLUSION

The project successfully demonstrates how AI and emotional intelligence can be integrated into digital music platforms. By detecting emotions from speech and leveraging Spotify's extensive library, the system provides mood-aligned music recommendations, enhancing user satisfaction and mental well-being. Future improvements could include multi-modal emotion detection and real-time streaming support.

12. FUTURE SCOPE

To enhance the accuracy, adaptability, and user experience of the emotion-based music recommendation system, the following future directions are proposed:

a. Refined Mood Mapping: Introduce a more nuanced mood map by associating each emotion with multiple relevant music genres, allowing for a richer and more tailored listening experience.
b. Enhanced Track Retrieval Strategy: Improve the recommendation quality by using combined emotion and genre keywords (e.g., “happy upbeat pop” or “sad acoustic ballads”) when querying the Spotify API, ensuring better alignment between emotional state and recommended music.
c. Multimodal Emotion Recognition: Incorporate additional modalities such as facial expression analysis or text-based sentiment analysis (e.g., from chat transcripts) alongside speech input to improve the accuracy and contextual understanding of emotional states.
d. Integration of Pretrained Audio Models: Leverage state-of-the-art pretrained deep learning models like Wav2Vec 2.0, YAMNet, or Whisper to extract high-level audio embeddings, which can outperform traditional handcrafted features like MFCCs in emotion recognition tasks.
e. Dataset Expansion and Diversity: Use or collect datasets featuring greater speaker diversity across genders, ages, languages, and accents, recorded under varying acoustic conditions, to enhance model generalization and performance in real-world scenarios.

13. REFERENCES

[1] Picard, R. W. (1997). Affective Computing. MIT Press. A foundational book introducing emotion-aware computing systems.

[2] Verma, G. K., & Tiwary, U. S. (2014). Multimodal emotion recognition using facial, speech and textual features. Multimedia Tools and Applications, 76(1), 4403–4425. https://doi.org/10.1007/s11042-016-3819-0

[3] Livingstone, S. R., & Russo, F. A. (2018). The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in North American English. PLoS ONE, 13(5), e0196391. https://doi.org/10.1371/journal.pone.0196391

[4] Hershey, S., Chaudhuri, S., Ellis, D. P. W., Gemmeke, J. F., Jansen, A., Moore, R. C & Wilson, K. (2017). CNN architectures for large-scale audio classification. In 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 131-135). IEEE. https://doi.org/10.1109/ICASSP.2017.7952132

[5] Spotify for Developers. (n.d.). Web API Reference. Retrieved April 21, 2025, from https://developer.spotify.com/documentation/web-api