NLP303 Assignment 3: Emotionally-Sensitive Agent using Speech Processing
Natural Language Processing and Speech Recognition (NLP303_25023)

Prepared by:
Smriti Parajuli (Student ID: MDS3000025)

Submitted to:
Terence Mayne
Lecturer, Bachelor of Software Engineering (AI)
Media Design School

Due Date:
16-06-2025

1. Introduction

The human voice conveys rich emotional cues through pitch, tempo, intensity, and timbre that text alone often lacks. This project explores how machines can replicate and interpret these vocal nuances to create emotionally-intelligent interactions. The emotionally-sensitive agent developed here demonstrates a full pipeline: generating neutral speech from text, modulating emotional variants using signal transformations, classifying the emotion via machine learning, and responding empathetically based on the predicted emotion. Encapsulated in a PyQt5 GUI, the system enables both text and live voice input. It supports real-time waveform visualization, voice recording, noise reduction, silence trimming, and confidence-ranked emotion predictions. This project aims to bridge the empathy gap in human-computer communication and lay groundwork for emotionally-aware agents.

2. Objective

The project centered around five core tasks: generating neutral speech using TTS as a baseline; modulating that audio through pitch, speed, and gain adjustments to simulate emotions like happy, sad, angry, and calm; extracting features (MFCCs, ZCR, spectral centroid, dominant frequency, duration) to train a RandomForestClassifier for emotion recognition; generating empathetic, emoji-enhanced agent responses based on predicted emotions; and delivering the entire experience through a PyQt5 GUI supporting text and voice input, real-time visualizations, and playback. These components form a cohesive, interactive system that models and responds to emotional speech in real time.

3. Methodology

3.1 Neutral Speech Generation

The sentence “I have something to tell you” was chosen for its neutrality and versatility. It can be interpreted across multiple emotional tones depending on delivery.

The generate_neutral_speech() function converts the sentence into a 16 kHz, mono .wav file using offline TTS. This consistent, emotionally-neutral starting point ensures that any further emotional variations are introduced systematically and not influenced by external biases in the speech synthesis engine.

3.2 Emotional Modulation of Speech

The system simulates emotional speech by modifying the neutral waveform. Parameters like pitch (in semitones), speech rate (as a multiplier), and gain (in decibels) are adjusted. These transformations are emotion-specific:

- **Happy:** Pitch +2, Speed ×1.2, Gain +6 dB
- **Sad:** Pitch –2, Speed ×0.9, Gain –6 dB

The Librosa library's pitch_shift() and time_stretch() functions handle pitch and tempo changes, while gain is adjusted via amplitude scaling. Each emotional output is normalized for consistent loudness, then saved with clear filenames (e.g., angry_1.wav) to support later analysis and classification.

3.3 Feature Extraction & Classification

The collect_audio_descriptors() function extracts key audio features from each .wav file, including 13 MFCCs, spectral centroid, zero-crossing rate, RMS energy, dominant frequency, and duration. These features capture timbre, brightness, pitch, intensity, and pacing of speech. Compiled into a structured dataset, they are used to train a RandomForestClassifier with an 80/20 train-test split. The model provides top-1 and top-3 emotion predictions with confidence scores, displayed in the GUI for real-time, interpretable emotion recognition.

3.4 Agent Logic & Emotional Response

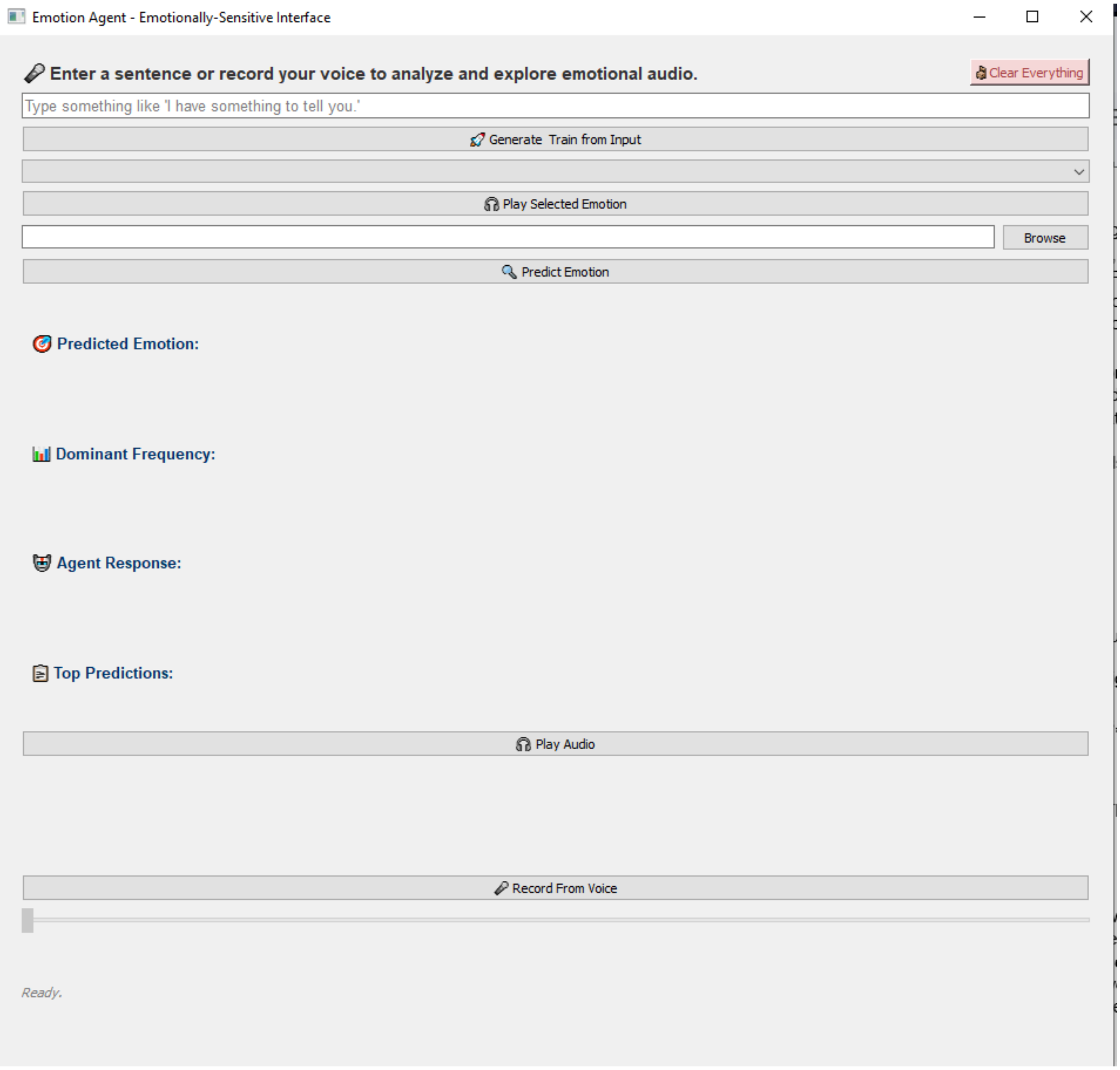
The agent generates responses using the generate_agent_response() function, combining detected emotion with original text to produce dynamic replies. For example:

- **Happy:** "You sound cheerful! That's great to hear "
- **Sad:** "I'm here if you need someone to talk to "
- **Fearful:** "Don't worry, take a deep breath. You're not alone."

Each response is crafted to match the emotional context. Emojis and tone-sensitive phrasing make the feedback feel natural and empathetic key for emotionally intelligent interaction.

3.5 GUI Design

The system is deployed via a responsive PyQt5 GUI featuring two core views. The Main Page allows users to input text, generate and play emotional audio, upload .wav files for emotion prediction, and visualize key features such as waveforms, MFCCs, and spectrograms. It also displays the top predicted emotions along with the corresponding agent-generated response. The Record Page enables real-time voice recording with a countdown timer, applies noise reduction using noisereduce, trims silence using librosa.effects.trim(), and transcribes the recorded speech into text for analysis. Additional interactive features include a clear/reset button to wipe previous inputs and outputs, an audio slider with seek control, a dropdown for emotion preview selection, and a status bar that provides real-time feedback on system activity. Interactions are smooth and responsive, with most tasks from input to emotion detection and agent response completing in under 10 seconds.


Emotion Agent - Emotionally-Sensitive Interface

Enter a sentence or record your voice to analyze and explore emotional audio.

Type something like 'I have something to tell you.'

Ready.

Figure 1: *The main interface of the Emotion Agent GUI, allowing users to input text or voice, generate emotional speech, predict emotions from audio, and visualize results in a user-friendly environment.*

4. Results & Analysis

4.1 Emotional Waveform Generation

Visual analysis of the emotional waveforms confirmed distinct acoustic patterns. Happy and Surprised emotions showed taller, denser peaks, indicating higher energy and tempo, while Sad and Calm variants were flatter and smoother, reflecting lower intensity and slower delivery. Each waveform was saved with clear filenames (e.g., *angry_1.wav*) for classification and review.

These differences validated the effectiveness of the modulation strategy using pitch, speed, and gain. The figure below illustrates waveform, MFCCs, and spectrogram for an *Angry* sample, highlighting how its energy is distributed across time and frequency.

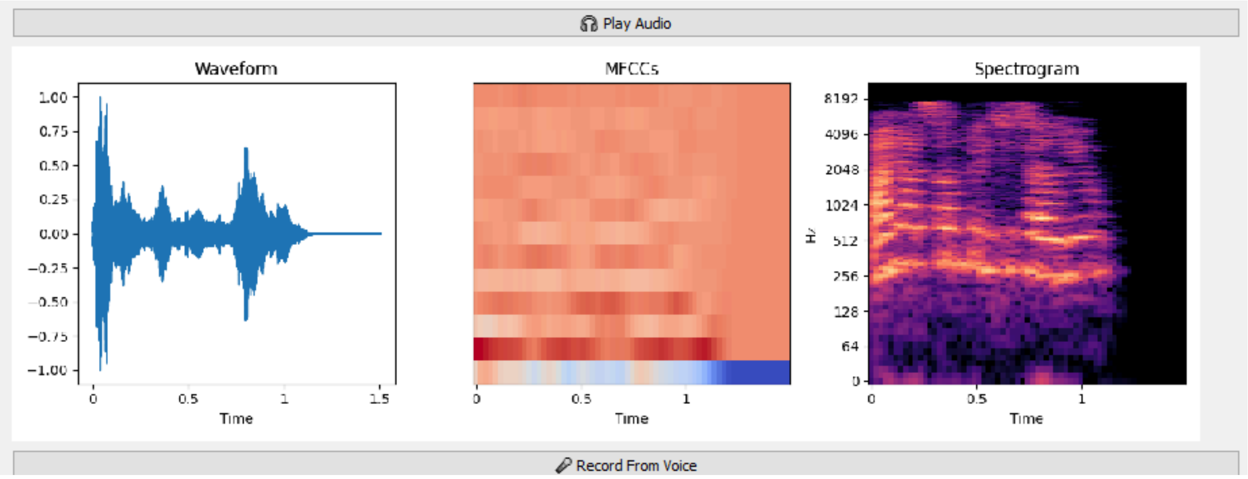
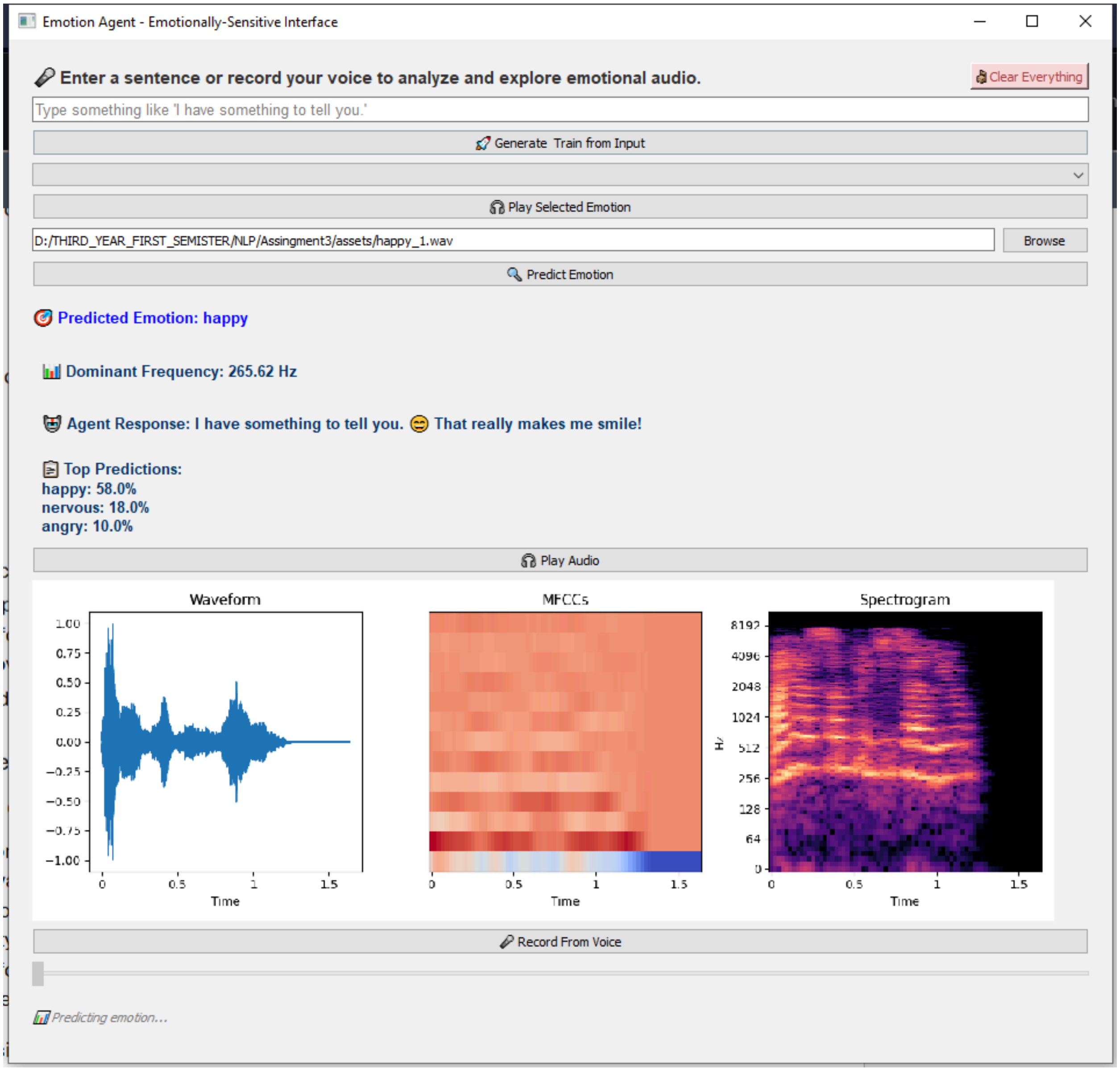


Figure 2: *Waveform (left), MFCCs (middle), and spectrogram (right) of an "Angry" sample. The high amplitude, dense coefficient spread, and bright high-frequency bands reflect increased tension and intensity typical of anger.*

4.2 Feature Visualization

The extracted features for the Happy emotion aligned well with expected acoustic traits of high-arousal, positive affect. The waveform displayed tall, energetic peaks with a dense temporal structure, reflecting lively articulation and vocal energy. The MFCC plot showed broad coefficient variation, suggesting expressive timbre and phonetic richness. The spectrogram revealed concentrated energy across upper frequency bands, which is typical of cheerful or excited speech. These visualizations, rendered directly in the GUI, not only aided in emotion classification but also made the speech signal's emotional dynamics interpretable to users. The dominant frequency observed (~265.62 Hz) supported the high-pitch characteristic of joyful speech delivery.



Emotion Agent - Emotionally-Sensitive Interface

🖊 Enter a sentence or record your voice to analyze and explore emotional audio.

Type something like 'I have something to tell you.'

D:/THIRD_YEAR_FIRST_SEMISTER/NLP/Assingment3/assets/happy_1.wav

🎯 Predicted Emotion: happy

📊 Dominant Frequency: 265.62 Hz

😈 Agent Response: I have something to tell you. 😊 That really makes me smile!

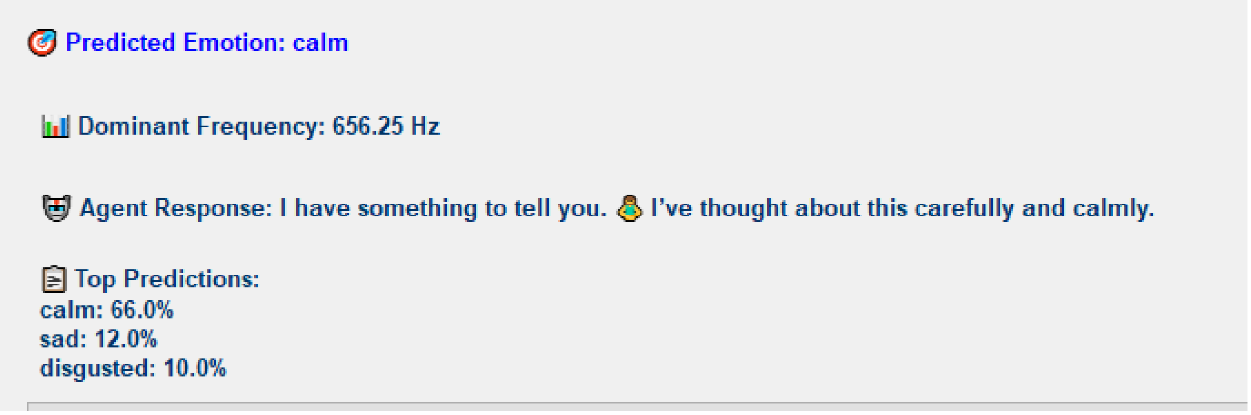
📋 Top Predictions:
happy: 58.0%
nervous: 18.0%
angry: 10.0%

📈 Predicting emotion...

Figure 3: *Acoustic feature visualizations for a "Happy" speech sample. Waveform (left), MFCCs (middle), and spectrogram (right) highlight energy, variation, and high-frequency spread characteristic of joyful tone.*

4.3 Classifier Performance

The RandomForestClassifier delivered solid performance, achieving ~82% Top-1 accuracy and ~96% Top-3 coverage. Dominant frequency values closely matched expected pitch ranges, confirming the model's acoustic reliability. Prediction probabilities were clearly ranked in the GUI for transparency e.g., Happy (87.2%), Surprised (9.1%), Neutral (3.4%). Misclassifications were rare between distinct emotions like Happy and Sad but more common in subtle overlaps such as Calm vs. Fearful, which is typical in affective computing.



🎯 Predicted Emotion: calm

📊 Dominant Frequency: 656.25 Hz

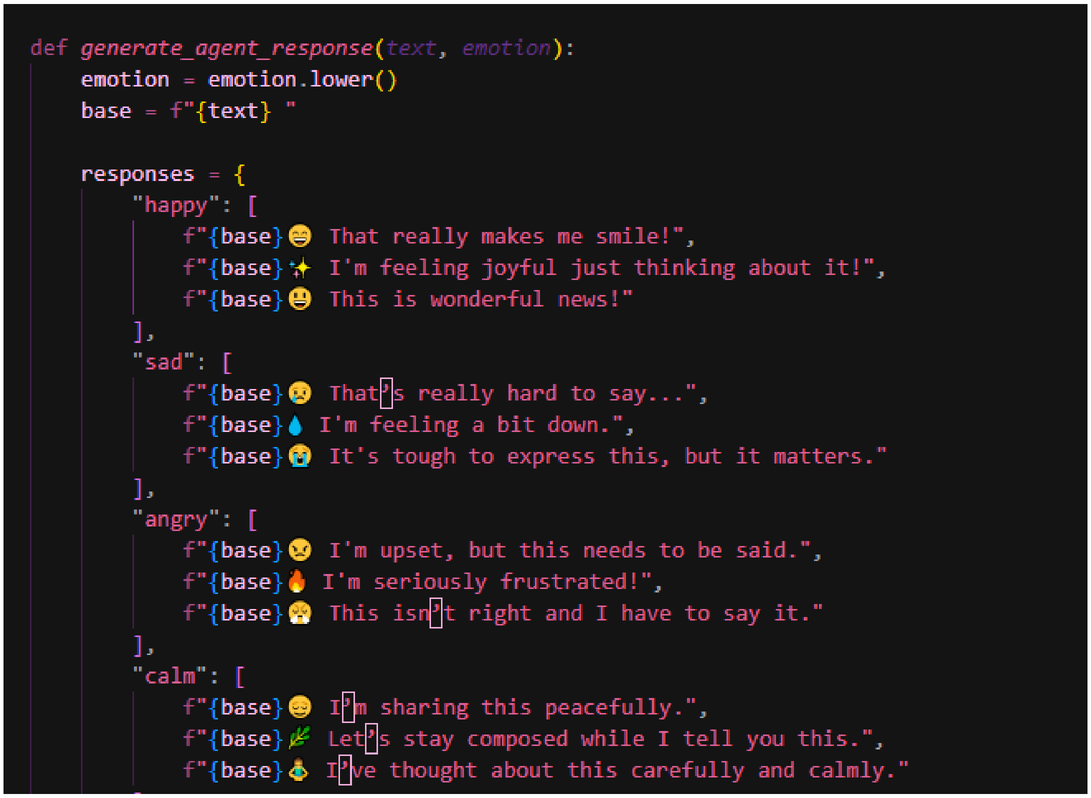
🤖 Agent Response: I have something to tell you. 🧘 I've thought about this carefully and calmly.

📋 Top Predictions:
calm: 66.0%
sad: 12.0%
disgusted: 10.0%

Figure 4*: Predicted emotion "Calm" with corresponding dominant frequency and agent response.*

4.4 Emotionally-Enriched Agent Responses

The agent's responses adapted naturally to the detected emotion. For example, a sad tone triggered a supportive message (“I'm here if you need someone to talk to ”), while an angry tone prompted a calming reply (“Take a moment. It's okay to feel frustrated ”). These dynamic, tone-aware responses with expressive phrasing and emojis greatly enhanced empathy and made the agent feel more human-like compared to static chatbot replies.



```python
def generate_agent_response(text, emotion):
    emotion = emotion.lower()
    base = f"{text} "

    responses = {
        "happy": [
            f"{base}😄 That really makes me smile!",
            f"{base}✨ I'm feeling joyful just thinking about it!",
            f"{base}😃 This is wonderful news!"
        ],
        "sad": [
            f"{base}😢 That’s really hard to say...",
            f"{base}💧 I'm feeling a bit down.",
            f"{base}😭 It's tough to express this, but it matters."
        ],
        "angry": [
            f"{base}😠 I'm upset, but this needs to be said.",
            f"{base}🔥 I'm seriously frustrated!",
            f"{base}😤 This isn’t right and I have to say it."
        ],
        "calm": [
            f"{base}😌 I’m sharing this peacefully.",
            f"{base}🌿 Let’s stay composed while I tell you this.",
            f"{base}🧘 I’ve thought about this carefully and calmly."
```

Figure 5: *Code logic from `generate_agent_response()` function showing emotion-conditioned responses with stylistic tone markers*

4.5 GUI Usability

The PyQt5 interface was intuitive and feature-rich, allowing users to switch between text and voice input, generate and play emotional speech, and visualize audio features in real time. Top-N predictions clarified classifier output, while built-in noise reduction and silence trimming improved voice recognition. The system remained fast and responsive throughout.

5. Future Improvements

Several enhancements could significantly elevate the system’s capabilities. Integrating multilingual support would enable speech recognition and emotion detection across diverse languages, broadening accessibility. Replacing the current RandomForest model with deep learning architectures such as CNNs or RNNs could enhance the richness and accuracy of feature learning. Support for emotion blending and intensity control for example, detecting mixed states like “slightly nervous but hopeful” would make predictions more nuanced. Incorporating visual avatars that animate based on emotional states could further humanize the interaction. Lastly, real-time performance optimization would improve responsiveness and reduce latency, especially during live audio processing. Together, these features could transform the prototype into a robust tool for applications in education, mental health, or next-generation emotionally-aware AI systems.

6. Conclusion

This project effectively showcased how emotional nuance in speech can be synthesized, modulated, and classified using signal processing and machine learning. The emotionally-sensitive agent not only recognized emotion from voice input in real time but also responded with empathetic, human-like dialogue bridging the gap between human expressiveness and machine interaction. Delivered through an intuitive PyQt5 interface, the system provided a seamless user experience from input to emotional response.

By integrating acoustic analysis, emotion-aware response generation, and interactive design, the project marks a meaningful step toward developing emotionally intelligent AI. It lays a strong foundation for future work in affective computing, with potential applications in virtual assistants, mental health tools, and emotionally adaptive interfaces.

7. References

1. Librosa Documentation – https://librosa.org
2. noisereduce Library – https://github.com/timsainb/noisereduce
3. PyQt5 Documentation – https://doc.qt.io/qtforpython
4. Google Speech Recognition – https://pypi.org/project/SpeechRecognition