



AI MEETING SUMMARIZER

*“Automating Meeting Transcription,
Diarization, and Summarization”*

PRESENTED BY
Sonal Verma





Motivation

- Modern meetings generate ***hours of audio***, hard to review manually.
- Need for ***automated summarization*** to improve productivity.

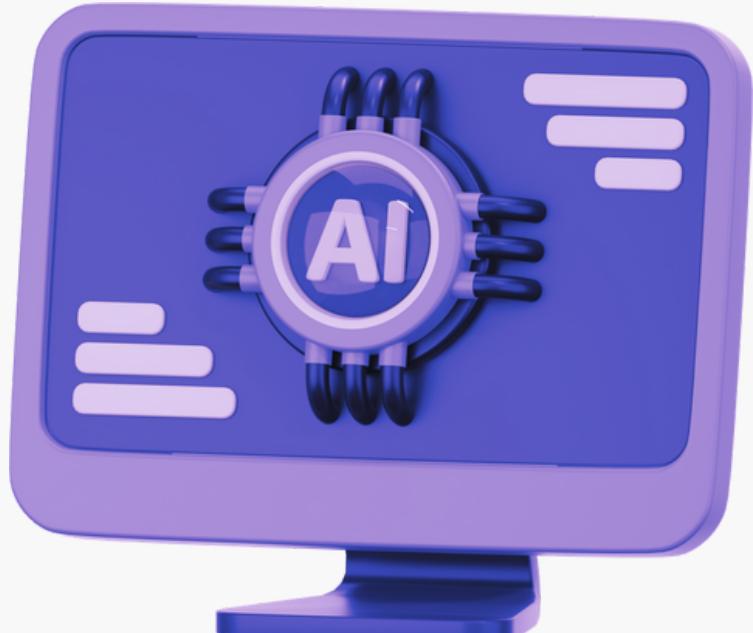
Challenges:

- Long recordings
- Multiple speakers
- Background noise
- Extracting key insights efficiently

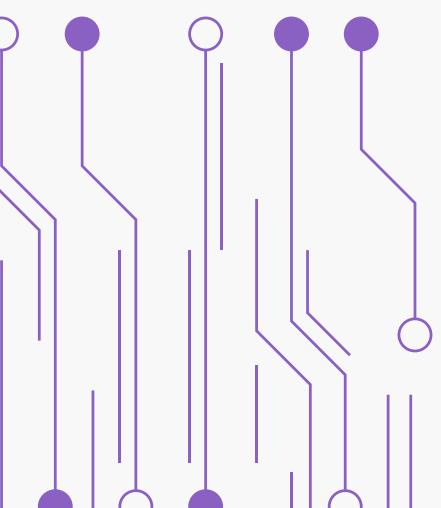
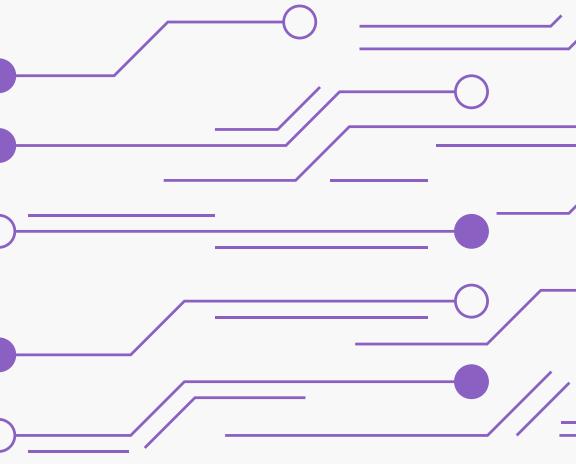




Project Overview

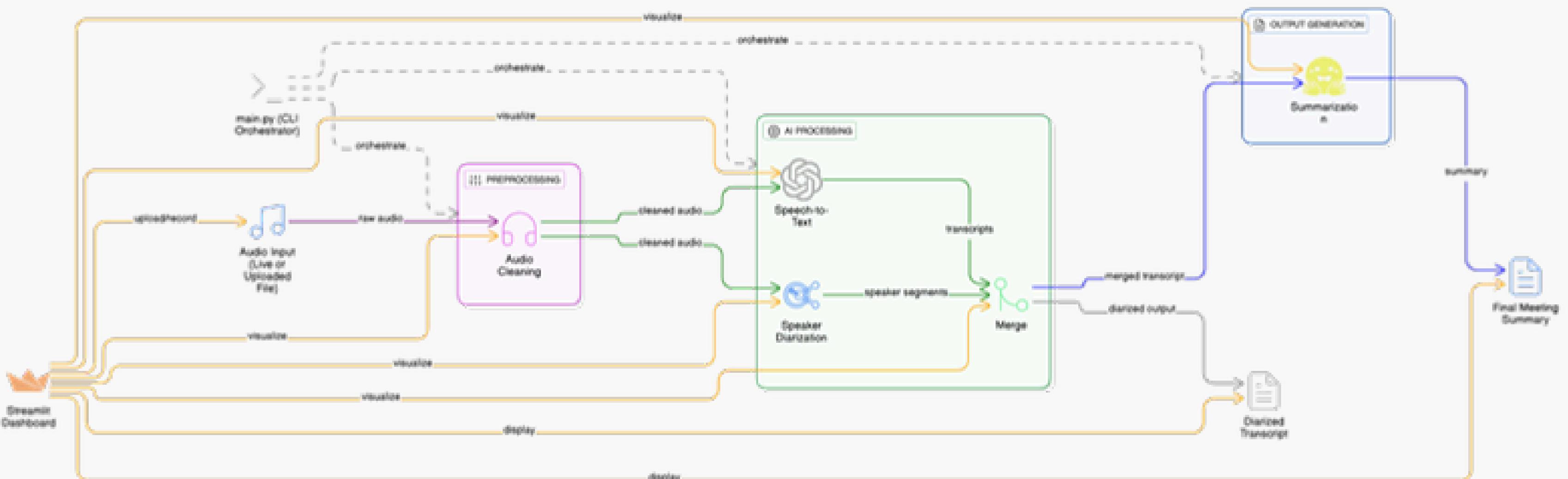


- 1 Cleans audio
- 2 Transcribes speech
- 3 Identifies speakers
- 4 Merges results
- 5 Generates AI summaries
- 6 Supports real-time or uploaded audio
- 7 Provides both CLI and Streamlit dashboard



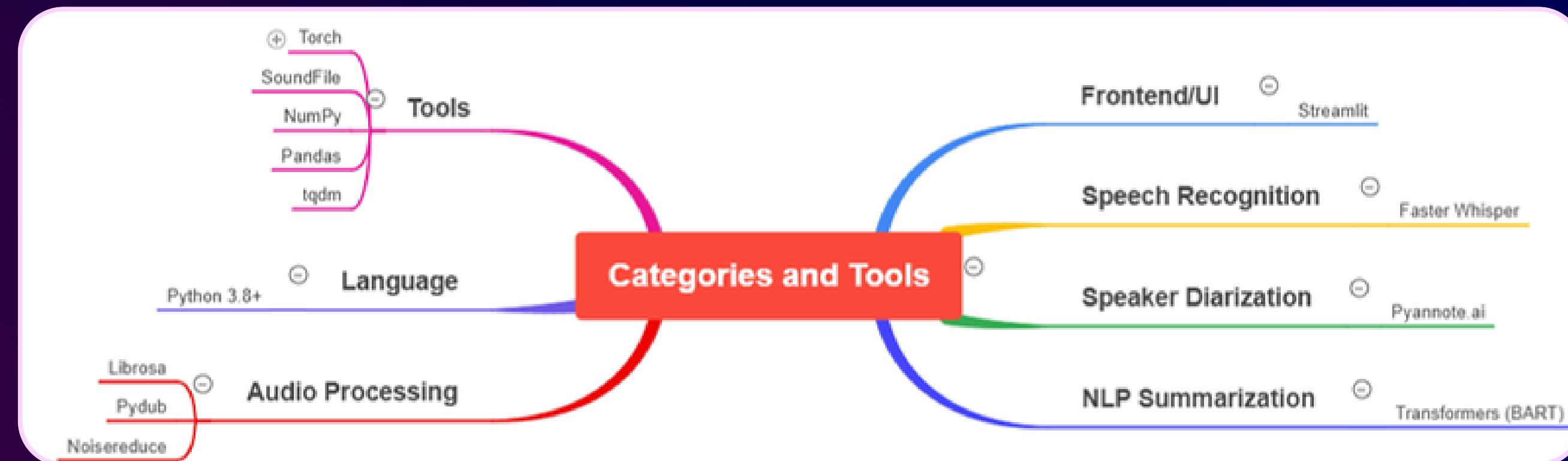
 MELTS

System Architecture



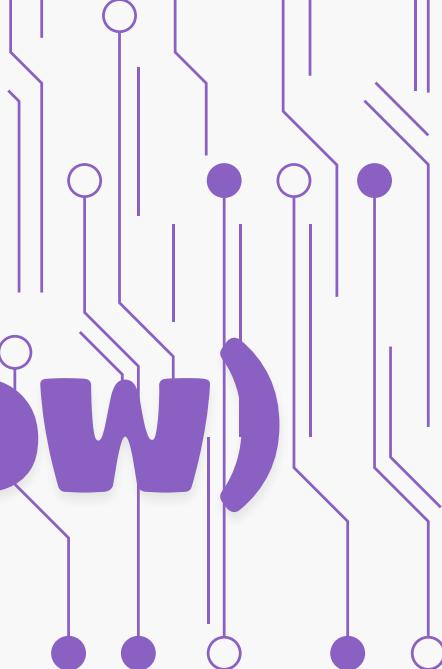


Tech Stack

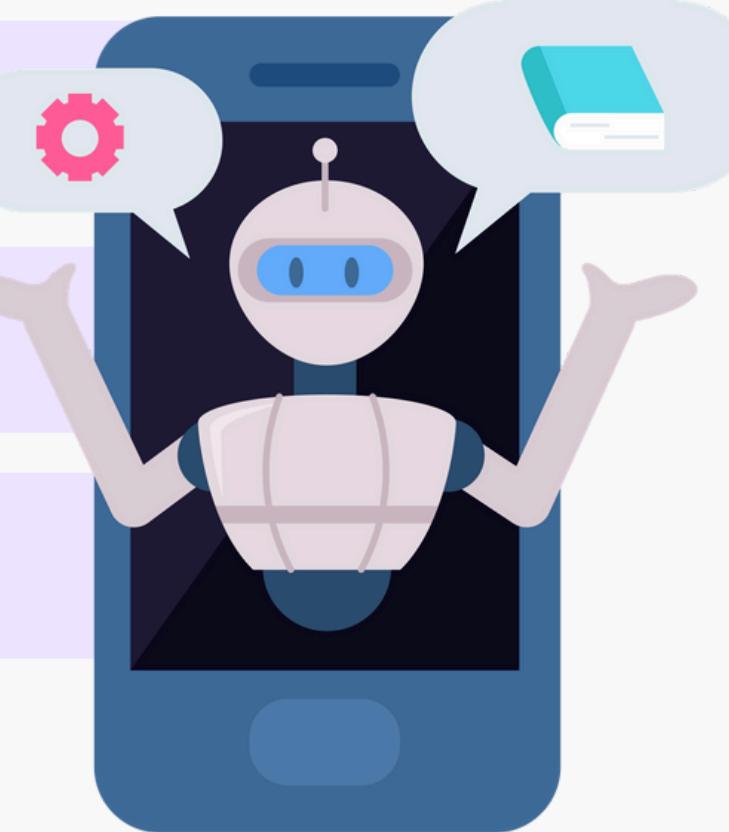




Pipeline Stages (Detailed Flow)



STEP	DESCRIPTION	SCRIPT/MODULE
1. Audio Cleaning	Noise reduction, resampling	milestone_1/audio_cleaner.py
2. Transcription	Faster Whisper model	milestone_2/usingfilemodel.py
3. Diarization	Speaker detection via Pyannote	milestone_4/dairization.py
4. Merging	Align speech & speakers	milestone_4/merge.py
5. Summarization	Transformer (BART)	milestone_4/summarizer.py





User Interface (Dashboard)

- Built with Streamlit
- Allows:
 - Uploading or recording audio
 - Running full pipeline
 - Viewing transcript, diarization, and summary

The screenshot shows a Streamlit-based dashboard titled "AI Speech-to-Text Dashboard". At the top right is a "Deploy" button. Below the title is a subtitle: "Record or upload audio → Transcribe, Diarize, and Summarize meetings effortlessly". The interface is divided into two main sections: "Input Options" on the left and "Output Results" on the right.

Input Options: A section for selecting input mode. It includes a radio button for "Live Recording" (selected) and a radio button for "Upload Audio File". A note below says "Please save this file to use it later for transcription." A button labeled "Click on 🎙 to start recording" with a microphone icon is shown, along with a timer indicating 00:00.

Output Results: A section for output results. It includes a note: "you can copy text by selecting and copying by pressing ⌘ Ctrl+C". Below this are three tabs: "Transcription" (selected), "Diarized Transcription", and "Summarized Notes". A "Raw Transcription" section contains the placeholder text "Transcription will appear here...".



Core AI Models

1. Whisper (Speech-to-Text)

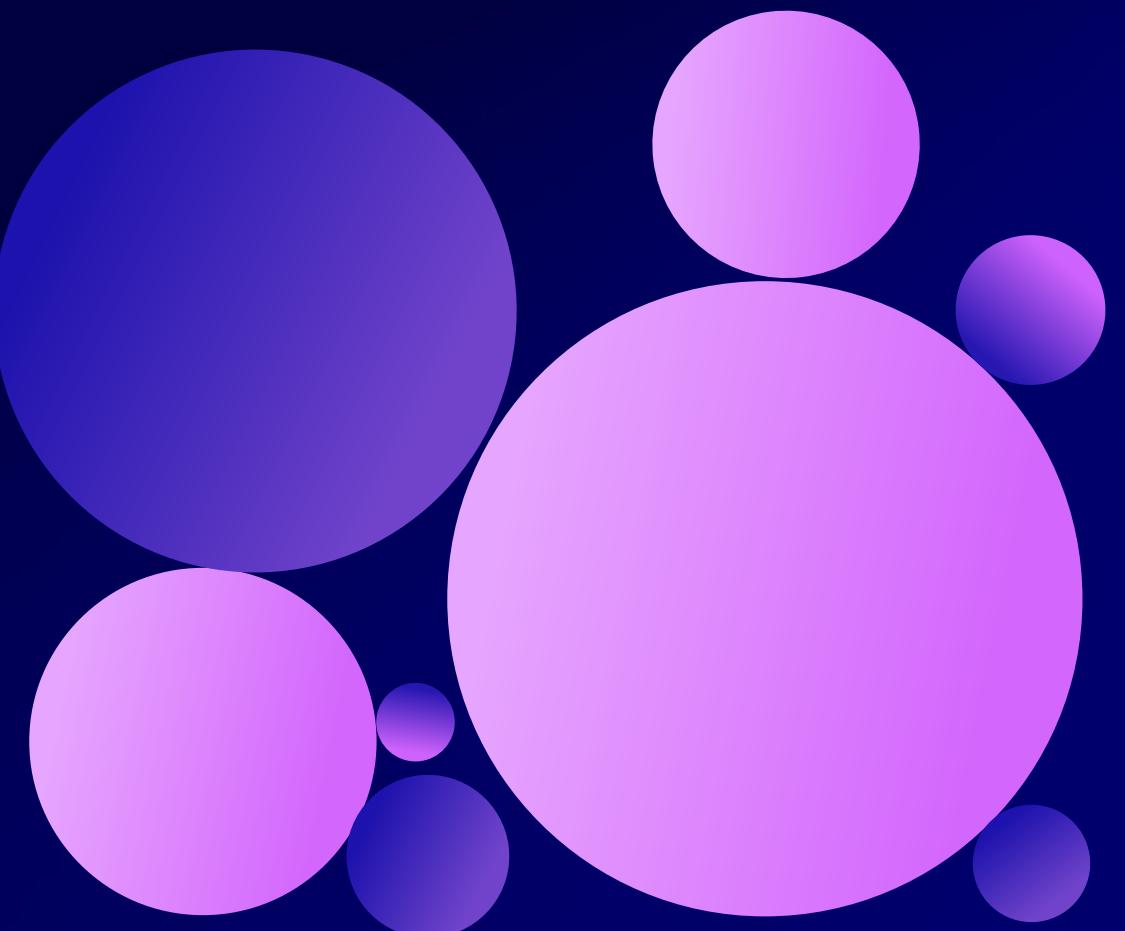
- Model: faster-whisper/small.en
- Converts speech → text with timestamps

2. Pyannote.ai (Diarization)

- Identifies who spoke when
- Cloud-based API

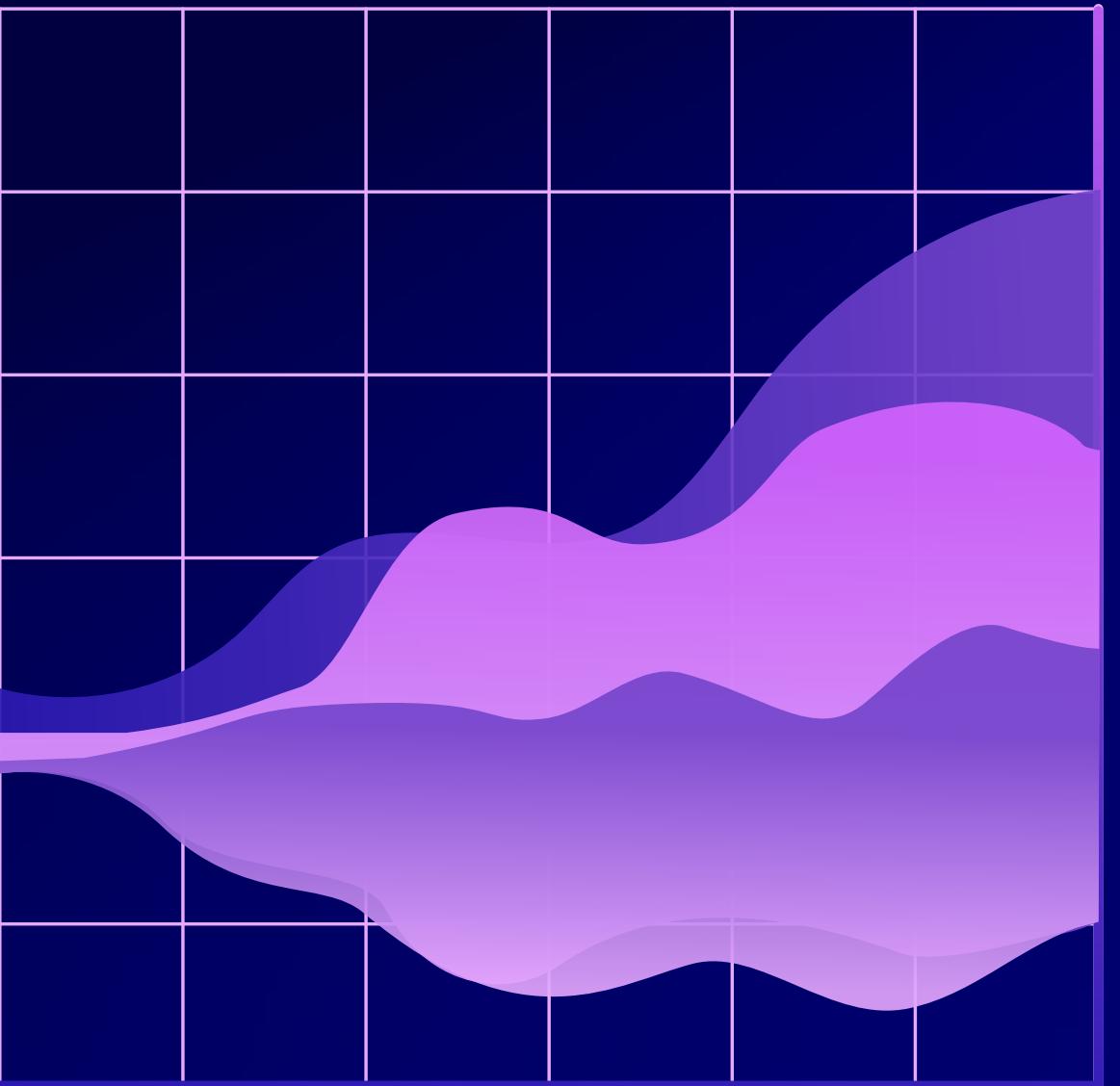
3. BART (Summarization)

- Model: facebook/bart-large-cnn
- Converts long transcripts into concise summaries



Evaluation & Testing

- WER (Word Error Rate): measured using jiwer
- Quality checks:
 - Clean waveform (manual)
 - Transcript accuracy
 - Speaker correctness
 - Summary coherence
- Independent testing of each milestone (modular design)





Key Features & Highlights

- Real-time and file-based audio support
- Accurate transcription with Whisper
- Multi-speaker diarization
- Summarization using Transformer models
- Modular design — each milestone testable
- Interactive Streamlit dashboard





Future Enhancements & Conclusion

- **Future Work::**

- Live meeting integration (Zoom, MS Teams APIs)
- Real-time diarization
- Multi-language support
- Sentiment and action item detection

- **Conclusion::**

- “AI Meeting Summarizer reduces hours of meeting analysis to minutes – enabling smarter collaboration and faster decision-making.”



Thank You

