AI-DRIVEN MULTI-MODEL ASSESSMENT FOR TEACHER PERFORMANCE ANALYSIS THROUGH AUDIO

IT3811-PROJECT WORK

Submitted by

MANOJ KUMAR P 311521205029
PRIYADHARSHINI V 311521205040
TANUJHAA G 311521205053

submitted to the Faculty of

INFORMATION TECHNOLOGY

in partial fulfillment for the award of the degree
of

BACHELOR OF TECHNOLOGY
in
INFORMATION TECHNOLOGY

MEENAKSHI SUNDARARAJAN ENGINEERING COLLEGE,
KODAMBAKKAM
(An Autonomous Institution)
ANNA UNIVERSITY::CHENNAI 600025

APRIL/MAY 2025

MEENAKSHI SUNDARARAJAN ENGINEERING COLLEGE, KODAMBAKKAM

(An Autonomous Institution)

AFFILIATED TO ANNA UNIVERSITY

BONAFIDE CERTIFICATE

Certified that this project report titled **AI-DRIVEN MULTI-MODEL TRANSFORMER ASSESSMENT FOR TEACHER PERFORMANCE ANALYSIS SYSTEM THROUGH AUDIO** is the bonafide work of **MANOJKUMAR P (311521205029),PRIYADHARSHINI V(311521205040), TANUJHAA G(311521205053)** who carried out project work under my supervision.

SIGNATURE

MRS. D. GAYATHRI M.E.
SUPERVISOR
ASSISTANT PROFESSOR
DEPARTMENT OF IT, MSEC
(An Autonomous Institution)
KODAMBAKKAM
CHENNAI 600024

SIGNATURE

DR. A. KANIMOZHI M.E., Ph.D.
HEAD OF THE DEPARTMENT
DEPARTMENT OF IT, MSEC
(An Autonomous Institution)
KODAMBAKKAM
CHENNAI 600024

Submitted for the project viva voce held at Meenakshi Sundararajan Engineering College on _____

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ABSTRACT

The Teacher Performance Analysis System is an artificial intelligence-based solution that is intended to measure teaching performance objectively by audio analyzing of lectures in the classroom. This cutting-edge system makes use of sophisticated machine learning algorithms such as Whisper for speech-to-text, sentiment analysis transformers, and proprietary audio processing algorithms to determine three important performance indicators: speech clarity, emotional delivery, and student engagement potential. The system analyzes recorded lectures, providing detailed assessments with visual aids and actionable recommendations for improvement. The system integrates audio feature extraction (MFCCs) with natural language processing to give comprehensive feedback. Teachers get scorecards with detailed performance benchmarks, while administrators get data-driven insights to plan professional development. The web-based interface facilitates simple upload of audio recordings or direct capture within the browser, with results displayed through an easy-to-use dashboard. Automating performance assessment, the system minimizes subjectivity within educator evaluations and provides predictable, quantifiable feedback. The project shows how transformer models can be successfully combined with legacy signal processing approaches to form effective teaching tools. This solution can have possible applications in teacher training programs, classroom observation systems, and self-improvement tools for educators who want to improve their instructional delivery.

ACKNOWLEDGEMENT

We are immensely grateful for the opportunity to express our heartfelt thanks to the **LORD ALMIGHTY** for showering His infinite blessings, wisdom, and guidance throughout this project. His unwavering support has been a constant source of strength, helping us overcome all challenges.

We extend our deepest gratitude to our **Correspondent, Dr. K.S. Lakshmi**, and **our Secretary, Shri N. Sreekanth**, for their visionary leadership, providing the facilities and a conducive environment that greatly facilitated the successful completion of our project as part of our internship.

We are sincerely thankful to our **Principal, Prof. Dr. S.V. Saravanan** of **Meenakshi Sundararajan Engineering College** for his constant encouragement, support, and guidance. We also express our profound thanks to **Prof. Dr. A. Kanimozhi**, Head of the **Information Technology Department**, for her invaluable guidance, insightful feedback, encouragement, and support throughout the completion of this internship project.

Our heartfelt thanks go to **Mr. Mohan Raj Vijayan, Assistant Professor and Project Coordinator and Mrs. D. Gayathri, Assistant Professor and Internal Guide,** for their invaluable suggestions, patience, guidance, and unwavering support throughout the duration of the project.

Above all, we express our deepest sense of gratitude, respect, and reverence to our beloved **PARENTS** for their constant motivation, moral support, and encouragement throughout our lives.

TABLE OF CONTENTS

	ABSTRACT	iii
	LIST OF FIGURES	vii
1	**INTRODUCTION**	**1**
	1.1 PROBLEM STATEMENT	1
	1.2 OBJECTIVES	2
	1.3 SCOPE	2
	1.4 MOTIVATION	3
2	**EXISTING SYSTEM**	**4**
	2.1 LITERATURE SURVEY	4
3	**SYSTEM REQUIREMENTS**	**11**
	3.1 SOFTWARE REQUIREMENTS	11
	3.2 HARDWARE REQUIREMENTS	12
4	**SYSTEM DESIGN**	**14**
	4.1 ARCHITECTURE	15
	4.2 LIST OF MODULES	16
	4.2.1 AUDIO PROCESSING MODULE	16
	4.2.2 SPEECH TRANSCRIPTION MODULE	17
	4.2.3 SENTIMENT ANALYSIS MODULE	18
	4.2.4 PERFORMANCE SCORING MODULE	19
	4.2.5 VISUALIZATION AND REPORTING MODULE	20
	4.3 ALGORITHM	22
5	**IMPLEMENTATION AND RESULTS ANALYSIS**	**25**
	5.1 IMPLEMENTATION	25
	5.1.1 AUDIO UPLOAD AND ANALYSIS DASHBOARD	26
	5.1.2 TEACHER PERFORMANCE ANALYSIS RESULTS	26
	5.1.3 FINAL PERFORMANCE SCORE VISUALIZATION	27
	5.1.4 IMPROVEMENT SUGGESTIONS FOR TEACHING PERFORMANCE	28

5.1.5 TRANSCRIPTION OF LECTURE AUDIO 29
5.2 RESULT ANALYSIS 31
5.2.1 PROJECT-SPECIFIC MODEL ANALYSIS 31
5.2.2 WHISPER AND DISTILBERT FOR TEACHER AUDIO PERFORMANCE ANALYSIS 35

6 CONCLUSION AND FUTURE WORK 40
6.1 CONCLUSION 40
6.2 FUTURE WORK 40

APPENDIX 42

A PROGRAMMING CODE 42

REFERENCES 52

LIST OF FIGURES

4.1	ARCHITECTURE DIAGRAM	15
4.2	AUDIO PROCESSING MODULE	17
4.3	SPEECH TRANSCRIPTION MODULE	18
4.4	SENTIMENT ANALYSIS MODULE	19
4.5	PERFORMANCE SCORING MODULE	20
4.6	VISUALIZATION AND REPORTING MODULE	21
5.1	AUDIO UPLOAD AND ANALYSIS DASHBOARD	26
5.2	TEACHER PERFORMANCE ANALYSIS RESULTS	27
5.3	FINAL PERFORMANCE SCORE VISUALIZATION	28
5.4	IMPROVEMENT SUGGESTIONS FOR TEACHING PERFORMANCE	29
5.5	TRANSCRIPTION OF LECTURE AUDIO	30
5.6	SPEECH-TO-TEXT MODELS ACCURACY COMPARISON	32
5.7	SENTIMENT ANALYSIS MODELS ACCURACY COMPARISON	32
5.8	TEST ACCURACY OF DIFFERENT SPEECH TO TEXT MODELS	33
5.9	TEST ACCURACY OF SENTIMENT ANALYSIS MODELS	34
5.10	COMPARATIVE ANALYSIS OF SPEECH RECOGNITION MODEL	36
5.11	COMPARATIVE ANALYSIS OF SENTIMENT MODELS	37
5.12	SPEECH-TEXT MODELS FOR TEACHER PERFORMANCE	38
5.13	TEXT MODEL COMPARISON FOR TEACHER SENTIMENT PREDICTION	39

CHAPTER 1

INTRODUCTION

Good teaching involves ongoing evaluation of performance, but conventional observation techniques are subjective and time-consuming. This project is solving these problems by creating an AI-powered Teacher Performance Analysis System to automatically evaluate teaching quality from audio recordings. The system evaluates speech patterns, emotional tone, and potential for engagement with cutting-edge machine learning models. By giving objective quantifiables and specific improvement recommendations, it empowers teachers to better their own practices without recourse to human raters. Bilingual support (English/Tamil) ensures usability across different education contexts. The technical method to assessing teachers marks an important advance in educational analysis, providing scalable, uniform appraisal that can accompany established observation strategies without wasting teachers' and administrators' time.

1.1 PROBLEM STATEMENT

Existing teacher assessment systems have a number of limitations that this project seeks to overcome. Classic classroom observations are subjective, time-consuming, and tend to capture only fleeting glimpses of teaching practice. Human assessors can be biased and inconsistent, while teachers are given delayed feedback that is not specific, actionable, or informative. There is an urgent need for an objective, scalable solution that can offer instant, data-based feedback on teaching performance, pinpoint specific areas of improvement in speech delivery and student engagement, facilitate analysis in multiple languages, minimize administrative burden on school

leaders, and provide consistent evaluation standards across institutions. The lack of such tools causes unequal teacher growth and lost chances for pedagogical development. These issues are addressed by this project through creating an automated tool that processes lecture audio to provide complete performance feedback with visual outcomes and bilingual features, allowing more efficient and equitable evaluation of teachers.

1.2 OBJECTIVES

The key objectives of this project are:

- To create an AI-based system that automatically assesses teaching effectiveness using speech and audio analysis.
- To offer objective performance measures (speech clarity, engagement, sentiment) with actionable feedback to teachers.
- To fuse audio and text features using a Multimodal Transformer for effective performance analysis.
- To generate performance scores and provide actionable suggestions for teaching improvement.
- To improve teacher training programs with data-driven insights for ongoing professional development.

1.3 SCOPE

The Teacher Performance Analysis System is a web-based application that assesses teaching performance through audio analysis. It examines lecture recordings through a multi-model pipeline measuring speech

clarity, sentiment expression, and student engagement. Speech clarity is examined with MFCCs and a neural network, sentiment expression is analyzed using DistilBERT, and engagement is predicted through a multimodal neural network combining audio features and transcripts. The frontend, built using HTML, CSS, and JavaScript, enables users to record or upload audio files, while the Flask backend performs model inference, data processing, and feedback generation. The system offers objective performance feedback, with support for multiple audio formats (WAV, MP3, M4A) and real-time recording for convenient use and access.

1.4 MOTIVATION

Traditional teacher assessments are commonly based on rare classroom observations or self-reporting surveys, which are biased and non-actionable. This project solves these limitations by using AI to provide uniform, data-driven evaluations. The incentive comes from three important needs: equity, efficiency, and actionability. Numerous educators, particularly those in less-funded schools, have limited access to high-quality assessment mechanisms. This system equalizes access by offering an automated, easy-to-use device that any instructor can employ. Furthermore, manual assessments are tedious for administrators, while this solution automates the process, providing instant feedback. In addition to scoring, the system makes targeted improvement recommendations—like modifying speech rhythm or modulating voice tone—allowing teachers to identify areas for improvement. Through the integration of audio and text analysis, the tool simulates how students hear lectures, picking up on technical delivery as well as emotional connection. Finally, the project aims to improve teaching quality by arming teachers with objective, AI-enhanced self-assessment tools.

CHAPTER 2

EXISTING SYSTEM

The present platform is an artificial intelligence-driven instructor performance analysis software that assesses lecture recordings on speech clarity, sentiment analysis, and engagement factors. It supports pre-recorded audio files (.wav,.mp3) and transcribes and analyzes them with Whisper and DistilBERT, respectively. The platform computes a composite performance score, data visualizations, and tailored feedback to assist educators in optimizing teaching techniques. Built with accessibility in mind, it has an easy upload interface and automatic reporting, allowing teachers to self-mark without technical skills.

2.1 LITERATURE SURVEY

Yun. T, Lim H, Lee.J, Song. M (2024) Teacher-leading Multimodal Fusion for Emotion Recognition in Conversations: A Cross-Modal Distillation Approach. Proceedings of the International Conference on Computational Linguistics and Speech Processing.

This paper presents TelME, a novel multi-modal AI-based system designed for emotion recognition in dialogue-based scenarios, with a strong potential for application in educational environments such as automated teacher performance analysis. TelME introduces a cross-modal distillation framework in which a powerful text-based transformer model, acting as the ”teacher,” transfers knowledge to comparatively weaker audio and visual modalities, termed as ”students.” This structure ensures that emotion cues found in textual data can guide and strengthen the understanding of non-textual signals, leading to a richer, more balanced multimodal emotion recognition model.One of

TelME's core innovations lies in its use of attention-based shifting fusion, a mechanism that dynamically weighs and integrates emotional cues from all three modalities based on conversational context. This fusion approach enables the model to capture subtle emotional transitions and deliver more nuanced emotion recognition, which is crucial in assessing the communicative behavior of teachers.The system was rigorously tested using benchmark datasets such as MELD (Multimodal EmotionLines Dataset) and IEMOCAP (Interactive Emotional Dyadic Motion Capture). It achieved state-of-the-art F1-scores of 67.37 and 70.48, respectively, surpassing previous models in recognizing both dominant and minority emotional classes. Notably, TelME's performance marked a 3.52% improvement over strong text-only baselines, showcasing the power of integrated multimodal learning. These results position TelME as a promising tool for real-time, AI-powered performance evaluation in educational technology systems, particularly for emotion-based feedback on teacher interactions and student engagement.

Criss. C. J., Carreon A. C, Massey, C. C, Davis. A. (2025)The role of technology in performance feedback on teacher practice: A systematic review. International Journal of Professional Development, Learners and Learning, 7(1), e2501.

The paper conducts a systematic review of 24 studies exploring how technology is used to deliver performance feedback to teachers. It finds that technology mainly supports the training phase, with real-time feedback being most common. The review highlights that feedback systems using video analysis, digital coaching tools, and automated performance metrics are effective in supporting reflective teaching practices and enhancing pedagogical strategies. Additionally, mobile and cloud-based platforms allow for flexible and asynchronous access to feedback, increasing scalability. However, issues like limited maintenance measures, inconsistent intervention lengths, and outdated practices persist. The study also points out that technological feedback

tools often lack personalization and adaptability to individual teaching contexts. Moreover, there is a scarcity of longitudinal studies tracking the sustained impact of such interventions. The study recommends enhancing technology's integration into feedback systems to promote evidence-based practices (EBPs) in classrooms and improve teacher performance. It suggests the use of AI-driven analytics, adaptive learning technologies, and multimodal feedback to overcome current limitations and create a more dynamic and supportive evaluation ecosystem for educators.

Lee. U, Jeong.Y., Koh. J, Byun. G, Lee. Y, et al. (2024)I See You: Teacher Analytics with GPT-4 Vision-Powered Observational Assessment. Smart Learning Environments, 11:48.

The study develops VidAAS, a video-based automated assessment system using GPT-4 Vision to analyze classroom behavior. VidAAS offers real-time, multimodal feedback on teachers' psychomotor (behavioral) skills and supports reflective teaching practices. The system utilizes computer vision and natural language understanding to monitor classroom dynamics such as teacher movement, gestures, board usage, and facial expressions, translating them into actionable insights for educators. It helps in reducing observer bias and offers continuous data-driven evaluations over time. It is highly accurate but needs improvements in assessing cognitive and emotional domains, such as identifying subtle student confusion or emotional disengagement. The paper emphasizes the need for better integration of affective computing to fully capture the instructional atmosphere. SWOT analysis and usability tests show promise for integrating AI in teacher assessments, revealing strengths in automation and objectivity, weaknesses in emotional analysis depth, opportunities in personalized professional development, and threats related to privacy and ethical concerns. The study concludes that VidAAS is a strong step forward in leveraging GPT-4 Vision for transforming how teacher performance is monitored and improved in educational environments.

Hu. J, Huang. Z, Li, J, Xu, L Zou. Y(2024) Real-Time Classroom Behavior Analysis for Enhanced Engineering Education: An AI-Assisted Approach. International Journal of Computational Intelligence Systems, 17:167.

This paper proposes an AI-driven system for real-time classroom behavior analysis, specifically tailored for engineering education environments. The system leverages an Emotion Recognition and Activity Monitoring (ERAM) model combined with activity tracking tools such as mouse and keyboard usage to provide a comprehensive view of student engagement during lab sessions and lectures. It operates continuously to detect variations in student focus, participation levels, and emotional responses, which are key indicators of learning efficacy. By analyzing these behavioral patterns in real-time, the system assists instructors in adapting their teaching strategies on-the-fly, offering interventions like targeted questioning or pace adjustments when disengagement is detected. This proactive approach is shown to significantly enhance student-instructor interaction and boost attentiveness. Experimental results demonstrate that classrooms implementing this AI-assisted system experienced an 8.44% improvement in teaching effectiveness compared to traditional non-AI environments. The study highlights the system's capability to bridge the gap between student behavior and teacher response through automated, intelligent insights, paving the way for smarter, more responsive educational experiences in STEM disciplines.

Bhavya.B, Neeraz. N, Parween. S, Bagawade.J. A, Kumar S, Swarnalatha S. R (2024)Job Performance of College Teachers in Higher Education with Reference to ICT. Journal of Informatics Education and Research, 4(2).

This paper comprehensively evaluates the role of Information and Communication Technology (ICT) in influencing the job performance of college-level educators. It utilizes the Technological Pedagogical Content Knowledge (TPACK) framework to analyze how ICT integration contributes to teaching effectiveness, student engagement, and the continuous professional

development of instructors. The findings reveal that strategic use of digital tools — such as smart boards, learning management systems (LMS), and data-driven performance tracking — significantly improves educators' ability to plan, deliver, and assess instruction. Moreover, the paper highlights how ICT fosters interactive and student-centered learning environments, empowering teachers to diversify their instructional strategies. However, the study also sheds light on several persistent challenges impeding the full realization of ICT benefits. These include limited access to updated technological resources, insufficient training opportunities, institutional resistance to pedagogical innovation, and a lack of administrative support. The paper strongly advocates for structured and targeted professional development programs aimed at equipping educators with practical ICT skills. By addressing these barriers and strengthening digital literacy among faculty, the paper suggests that educational institutions can unlock the full potential of ICT in enhancing both teaching practices and student learning outcomes.

Dimitriadou. E, Lanitis. A (2023)An Integrated Framework for Developing and Evaluating an Automated Lecture Style Assessment System.Education and Information Technologies,10(5)

This study presents a comprehensive and integrated framework for the development and evaluation of an automated lecture style assessment system. The system is designed to assess the quality of lecture delivery using a variety of biometric and behavioral features extracted from classroom videos. These features include facial expressions, hand and body movements, speech rate, and overall physical activity during the lecture, all of which contribute to a more holistic understanding of the instructor's presentation style.The system offers both real-time and cumulative evaluations of lecture quality by analyzing these features through sophisticated video and audio processing algorithms. Its real-time capability enables educators to receive instantaneous feedback, while the cumulative scoring provides a broader overview of lecture performance,

making it highly beneficial for continuous professional development.What sets this model apart from previous systems is its use of a stakeholder-informed feature selection process. This approach ensures that the chosen parameters for evaluation are not only technically significant but also practically relevant to educators and institutional goals. As a result, the system achieves a high level of reliability and aligns better with human assessment standards.Empirical evaluation demonstrated that the automated system performs on par with, and in many cases surpasses, human raters in assessing lecture quality. This outcome validates the model's potential as an objective and scalable tool for lecture assessment in higher education. By offering a standardized and efficient method of evaluating teaching delivery, the proposed framework contributes significantly to the advancement of AI-driven educational technologies.

Ordóñez-Ávila R, Salgado Reyes.N, Meza.J, Ventura. S(2023)Data mining techniques for predicting teacher evaluation in higher education: A systematic literature review. Heliyon, 9, e13939.

This paper presents a systematic literature review focused on the application of data mining techniques to predict teacher evaluation outcomes based on student-related data. The authors examine a comprehensive range of studies that utilize advanced computational methods to improve the objectivity and reliability of faculty performance assessments in higher education.The review highlights several prominent techniques employed across various studies, including decision trees, support vector machines (SVM), artificial neural networks (ANN), and fuzzy logic systems. Among these, fuzzy logic stands out as particularly effective due to its capacity to handle imprecise and subjective human responses, making it highly suitable for evaluating teaching effectiveness—a domain inherently influenced by qualitative factors such as student perceptions and learning environments.Furthermore, the paper emphasizes the growing role of Educational Data Mining (EDM) as an innovative and data-driven approach in the academic sector. EDM enables

institutions to derive meaningful insights from vast amounts of educational data, thereby facilitating more evidence-based decision-making in teacher evaluations. This shift toward analytics-based assessments marks a significant step in enhancing transparency and consistency in performance reviews, reducing potential biases inherent in traditional feedback mechanisms.The authors also point out challenges in implementing EDM approaches, such as data privacy concerns, heterogeneity of data sources, and the need for domain-specific customization of algorithms. Nonetheless, the review underscores that the integration of these intelligent systems holds strong promise for the future of educational quality assurance by supporting more nuanced and data-informed evaluations of teaching practices.

CHAPTER 3

SYSTEM REQUIREMENTS

3.1 SOFTWARE REQUIREMENTS

PROGRAMMING LANGUAGE

Python 3: Python is the primary programming language used for this project due to its readability, ease of use, and extensive ecosystem.

LIBRARIES AND TOOLS

- **OS**: Used for handling file operations, like saving the uploaded audio file temporarily.
- **torch (PyTorch)**:Used to set the number of CPU threads for efficient performance and to select between CPU and GPU for model operations.
- **soundfile (sf)**: Used for reading the uploaded audio file in .wav format.
- **numpy (np):**:Used for numerical operations, such as generating random engagement scores and calculating averages.
- **flask**:Used to create the REST API server which listens for audio file uploads and returns analysis results.

- **NLTK/spaCy**:These NLP libraries assist in text preprocessing, tokenization, and entity recognition, enhancing the system's ability to interpret and analyze resume content accurately.

DEVELOPMENT TOOLS AND ENVIRONMENT

- **Flask Server**: Hosts the API endpoints (/ and /analyze) and handles the backend logic.
- **Localhost Setup**: The backend runs on localhost (probably at port 5000) and accepts requests from the frontend running on localhost:3000.
- **GPU/CPU Auto Selection:**: The code automatically uses GPU (if available) for faster processing or falls back to CPU.

3.2 HARDWARE REQUIREMENTS

- **Processor:** A multi-core processor, such as an Intel Core i5 or AMD Ryzen 5, is recommended to handle machine learning model training and LLM inference efficiently. A higher-end CPU (e.g., Intel i7/i9 or Ryzen 7/9) improves performance for large-scale data processing.

- **Memory (RAM):** A minimum of 8GB RAM is required for basic operations, but 16GB or more is recommended to ensure smooth performance during concurrent ML model execution and LLM API calls. Larger RAM capacity reduces latency when processing multiple resumes simultaneously.

- **Storage:** An SSD with at least 50GB of free space is essential for storing resumes, trained models, and database files. SSDs offer faster read/write speeds compared to HDDs, significantly improving data retrieval times and application responsiveness.

- **Internet Connection:**A stable broadband connection is required for API interactions (e.g., Gemini-1.5) and cloud-based deployments. For local testing, a high-speed connection ensures uninterrupted data fetching and model updates.

CHAPTER 4

SYSTEM DESIGN

The architecture proposed is a sophisticated AI-driven system aimed at objectively assessing a teacher's performance by analyzing their lecture audio. The process starts at the Audio Input stage, where a recorded lecture by the teacher is input. This recording is then passed through a chain of Processing Models that execute several tasks. First, the audio is converted to text using the Whisper model, a robust speech 12 natural language interpreting agent. In the meantime, MFCC features are extracted from the audio signal, retaining important speech characteristics such as tone, pitch, and energy. These two streams of data—text and audio features—are subsequently fused by a Multimodal Transformer, a model that effectively combines information from multiple modalities to create an even richer representation of the input. Once the textual and audio features are fused, they are passed through Fully Connected (FC) Layers, which operate on the mixed data to produce quantifiable performance scores. They include a Speech Score, which measures the intelligibility, rate, fluency, and pronunciation of the teacher's speech; a Sentiment Score, which indicates the emotional tone and style of delivery; and an Engagement Score, which estimates how engaging and interactive the teacher's lecture was from vocal dynamics and language patterns. These individual scores are extracted at the Calculation phase, where they are averaged to calculate a Final Performance Score—one that encapsulates the quality and effectiveness of the lecture. The result is a Teacher Performance Report, which holds all the metrics judged in a well-organized form. The report can be used for self-improvement, institutional assessment, or professional growth, offering an equitable and data-driven assessment system. The entire system seamlessly integrates speech recognition, audio signal processing, and deep

learning models to give an end-to-end solution for performance evaluation in learning environments.

4.1 ARCHITECTURE

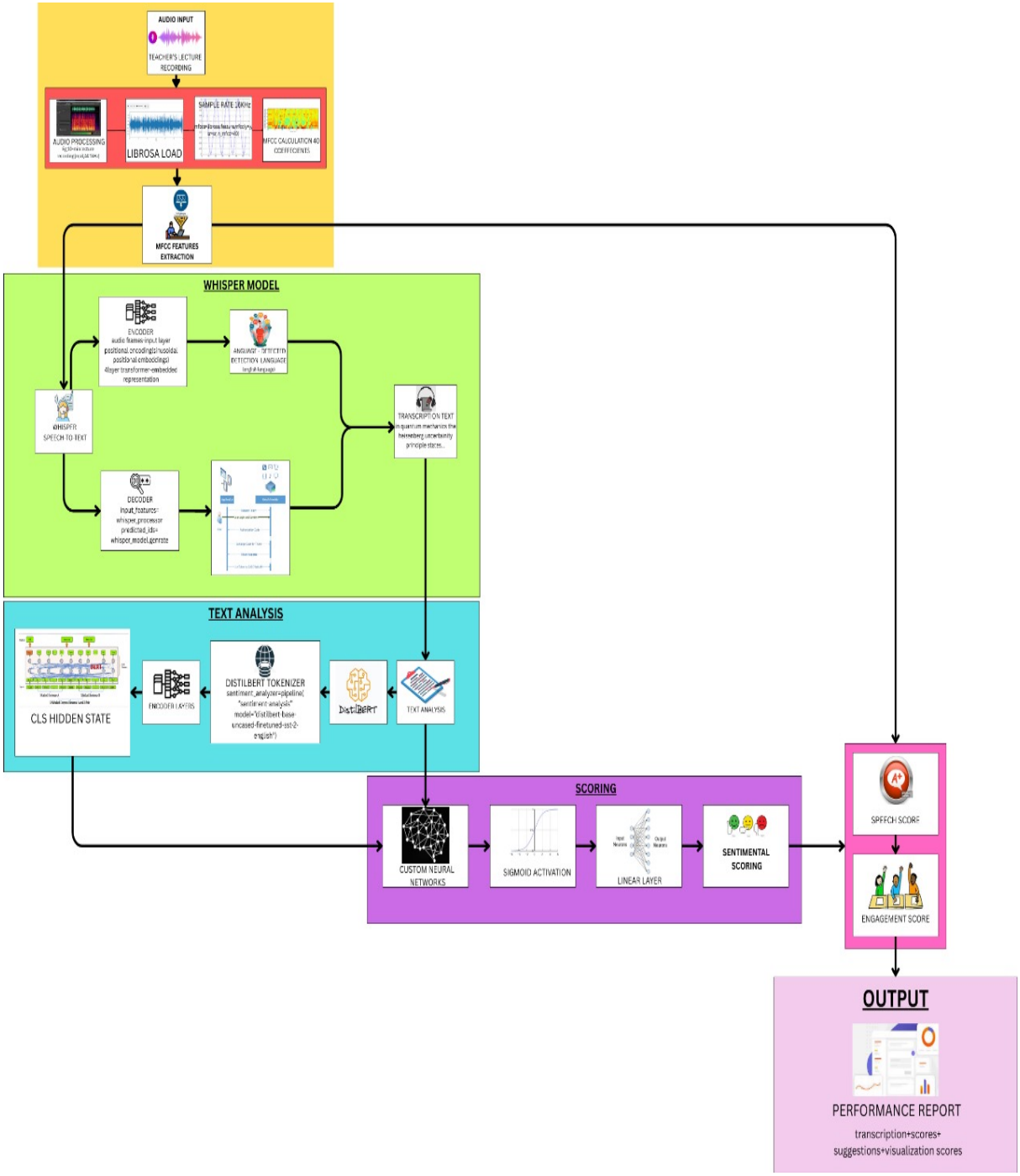

Figure 4.1 ARCHITECTURE DIAGRAM

4.2 LIST OF MODULES

The system has been broken down into relevant modules that perform different functions to improve usability and keep the code neat and clean. Below is the description of the main modules.

1. Audio Processing Module.
2. Speech Transcription Module.
3. Sentiment Analysis Module.
4. Performance Scoring Module.
5. Visualization Reporting Module.

4.2.1 AUDIO PROCESSING MODULE

The Audio Processing Module is intended to standardize and streamline lecture audio files using a structured workflow. The INPUT phase receives raw lecture audio, which is then processed under a PROCESS phase that includes a Format Check to check for compatibility and consistency. This process checks the format, bitrate, and other technical details of the audio and corrects any discrepancies if found. Lastly, the OUTPUT stage produces a Unified Audio file, normalized for easy playback, distribution, or additional processing. This module provides reliability and consistency in audio content, which is ideal for educational websites, archives, or automated transcription systems. The whole process increases efficiency and minimizes manual intervention.

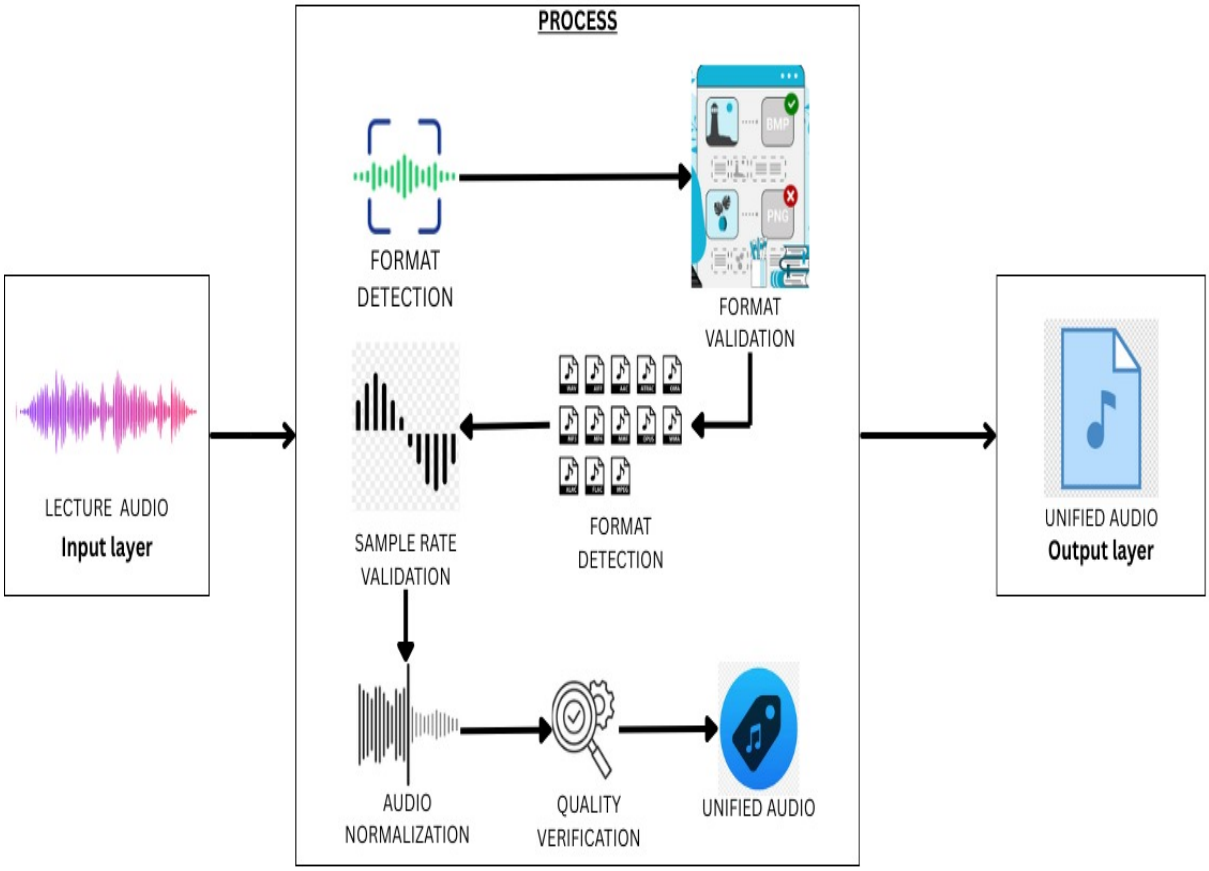

Figure 4.2 AUDIO PROCESSING MODULE

4.2.2 SPEECH TRANSCRIPTION MODULE

The Speech Transcription Module transcribes audio chunks into precise text transcripts through the Whisper Transcribe process. The INPUT is segmented audio chunks, which are fed into Whisper, a highly sophisticated AI-powered speech recognition system, to provide high accuracy in transcription despite different accents or background noise. The PROCESS step turns spoken language into written language, with context and clarity preserved. The OUTPUT provides a Text Transcript for subsequent use in documentation, subtitling, or analysis. This module is suitable for lectures, meetings, or any oral content needing textual representation, providing efficiency, scalability, and automated accuracy

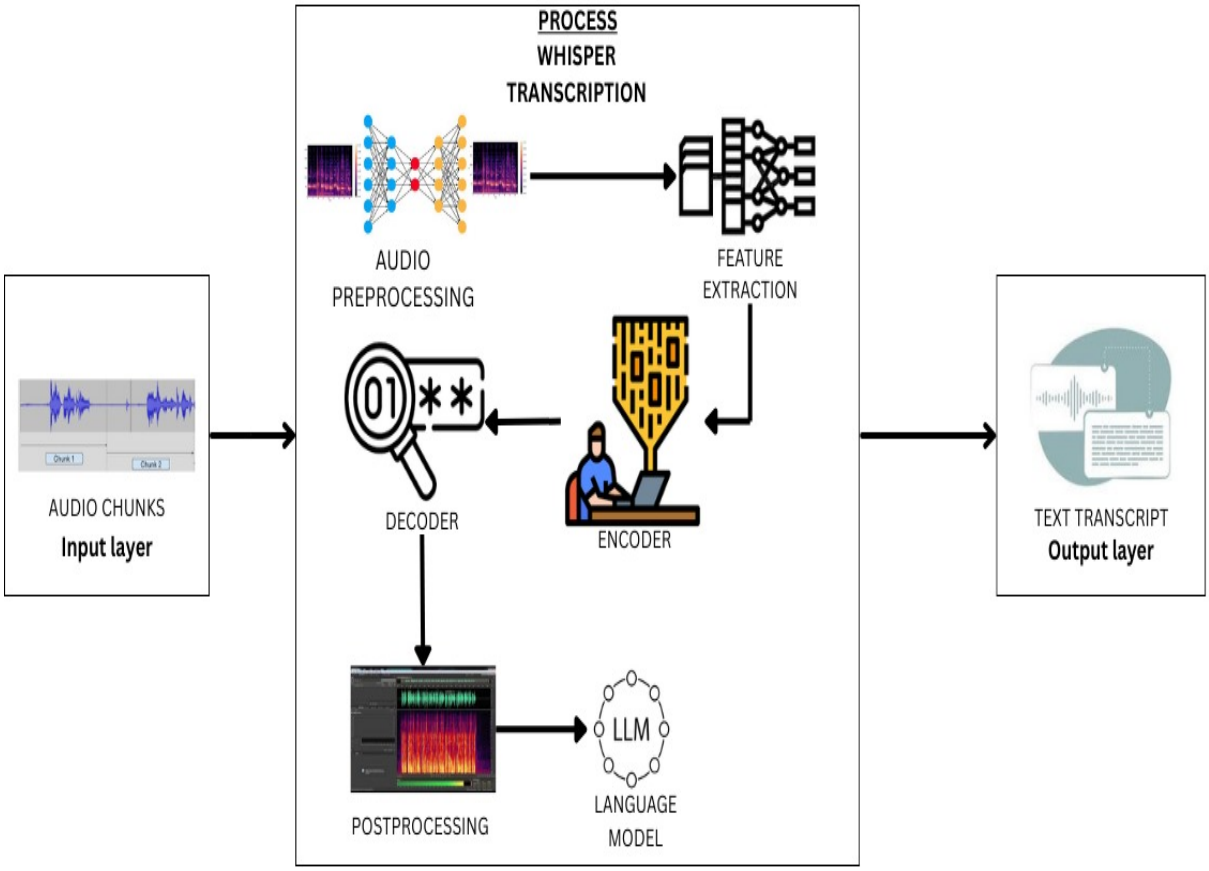

Figure 4.3 SPEECH TRANSCRIPTION MODULE

4.2.3 SENTIMENT ANALYSIS MODULE

The Sentiment Analysis Module analyzes the emotional tone of transcribed text through DistilBERT, a lightweight but effective NLP model. The INPUT is Transcript Text, which is analyzed through DistilBERT Analysis to identify sentiment subtleties—positive, negative, or neutral. The PROCESS phase utilizes contextual awareness to determine a Sentiment Score, measuring emotional undertones with high precision. The OUTPUT delivers actionable insights, which are helpful for feedback analysis, customer conversations, or content moderation. The module is efficient, scalable, and suitable for use cases demanding automated sentiment evaluation, for example, market research, educational feedback, or social media monitoring.

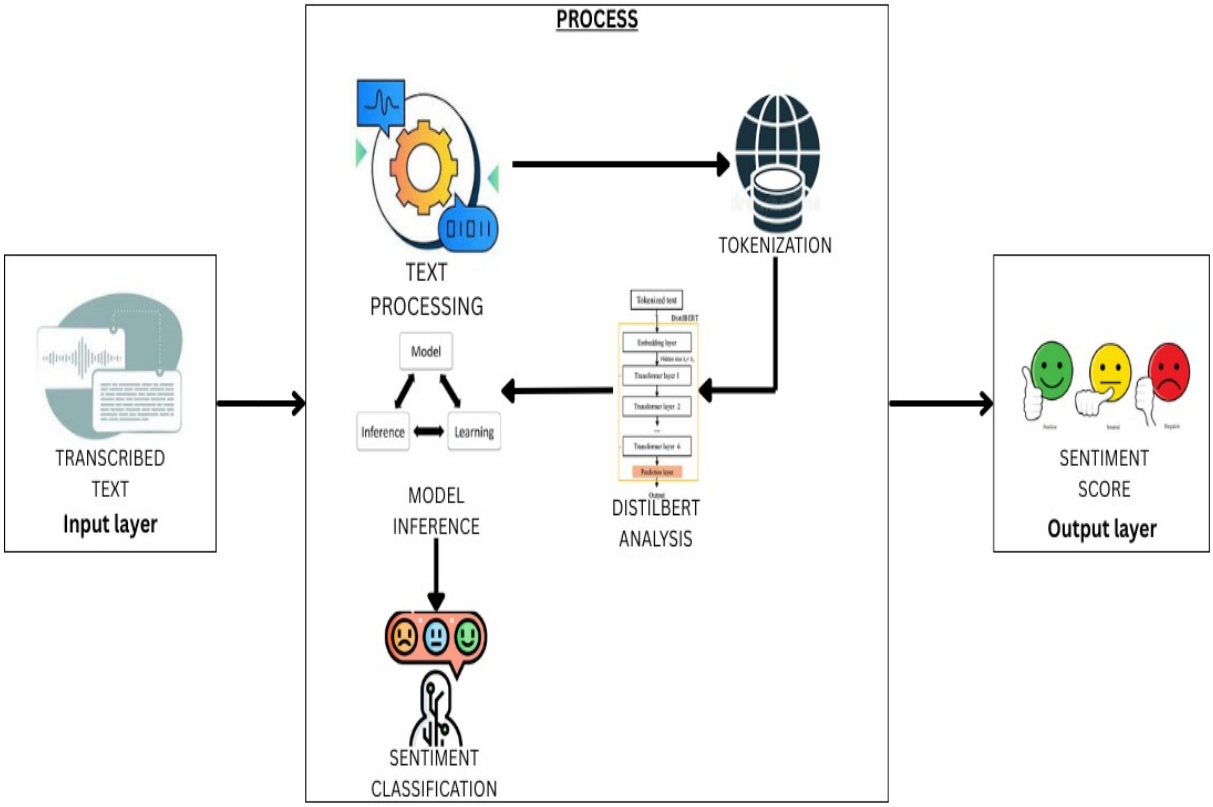

Figure 4.4 SENTIMENT ANALYSIS MODULE

4.2.4 PERFORMANCE SCORING MODULE

The Performance Scoring Module combines MFCC Features (acoustic features) and Sentiment Score (sentiment scores) to assess speaker performance holistically. The INPUT merges the multimodal inputs, which then go through Neural Fusion—a deep learning operation where audio and text-based insights are combined—before resulting in Score Computation for meaningful measures. The OUTPUT gives an overall overview with Speech quality, Engagement level, and Sentiment Score, presenting a complete performance analysis. Perfect for presentation, lecture, or public speaking analysis, this module allows data-driven feedback to improve communication effectiveness and audience impact.

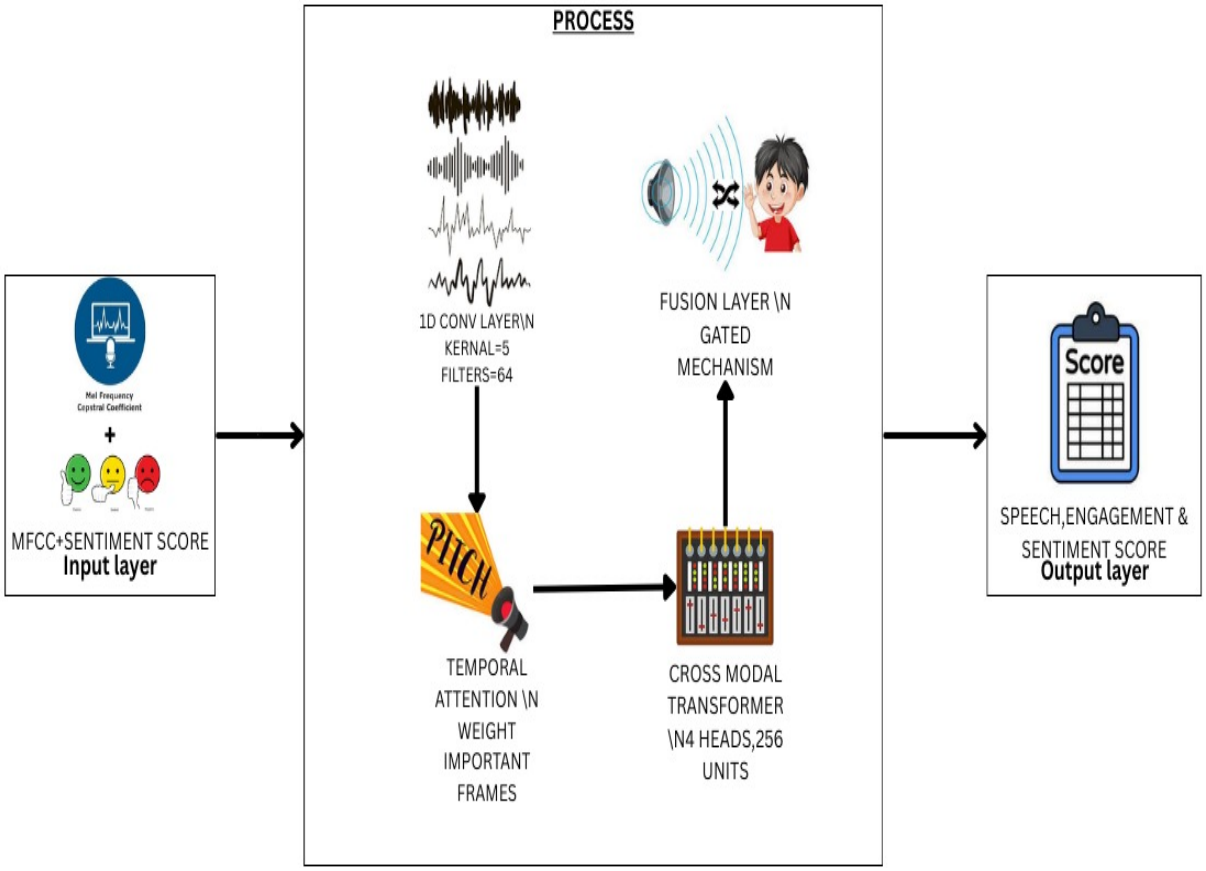

Figure 4.5 PERFORMANCE SCORING MODULE

4.2.5 VISUALIZATION AND REPORTING MODULE

The Visualization And Reporting Module converts raw performance measures into actionable information by processing All Scores and Transcript data. The INPUT integrates quantitative scores (e.g., engagement, sentiment) with text content, which are processed through Graph Plotting to provide intuitive visual representations. The OUTPUT produces an interactive Dashboard View with easy-to-understand trends, comparisons, and Suggestions for improvement. This module is perfect for instructors, presenters, or analysts who want to assess communication effectiveness, monitor progress, and make informed decisions. By aggregating intricate data into easy-to-use visuals, it improves comprehension and enables specific improvements.

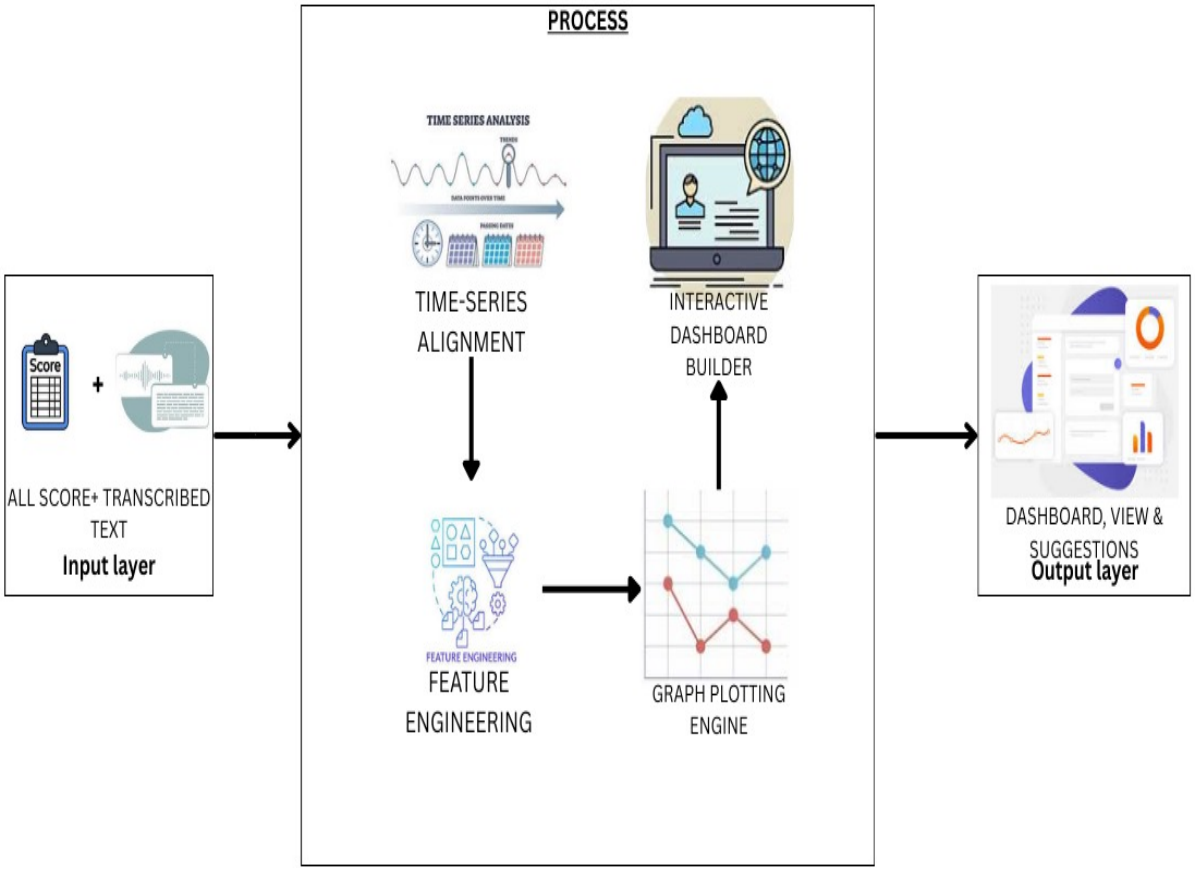

Figure 4.6 VISUALIZATION AND REPORTING MODULE

4.3 ALGORITHM

Algorithm 4.1 Whisper Speech Recognition Pipeline

Input: audio_path – path to audio file (WAV/MP3)
Output: transcription – text output, language – detected language

1: **Audio Preprocessing**
2: Load audio: $y, sr \leftarrow$ librosa.load(audio_path, sr=16000)
3: Normalize amplitude: $y \leftarrow y/\max(|y|)$
4: Pad/trim to 30s chunks if needed
5: **Feature Extraction**
6: Compute log-Mel spectrogram: $S \leftarrow$ WhisperFeatureExtractor(y)
7: $input\_features \leftarrow$ WhisperProcessor$(S, return_tensors = "pt")$
8: **Language Detection**
9: Generate language tokens: $lang\_tokens \leftarrow$ model.generate$(input_features, task = "detect")$
10: $language \leftarrow$ decode_language_id$(lang_tokens)$
11: **Transcription**
12: Force language or auto-detect: $forced\_ids \leftarrow$ language_to_token_id$(language)$
13: $predicted\_ids \leftarrow$ model.generate$(input_features, forced_decoder_ids = forced_ids)$
14: $transcription \leftarrow$ WhisperProcessor.batch_decode$(predicted_ids)$
15: **Post-processing**
16: Remove special tokens (¡—...—¿)
17: Capitalize sentences
18: Correct common ASR errors via lookup table
19: **Example:**
20: Audio: "Good morning class" $\rightarrow$ Transcription: "Good morning class", Language: "en"

Algorithm 4.2 DistilBERT Sentiment Analysis

Input: text – raw input text (max 512 tokens)

Output: sentiment_score – polarity value [-1,1], confidence – prediction certainty [0,1]

Text Preprocessing

Convert to lowercase

Remove URLs/special chars

Expand contractions ("don't" $\rightarrow$ "do not")

Tokenization

$tokens \leftarrow \text{DistilBertTokenizer}(\text{text}, \text{max\_length} = 512, \text{truncation} = True, \text{padding} =' max\_length')$

Model Inference

$outputs \leftarrow \text{DistilBertForSequenceClassification}(\text{input\_ids} = tokens['input\_ids'], \text{attention\_mask} = tokens['attention\_mask'])$

Score Calculation

$logits \leftarrow outputs.logits$

$probs \leftarrow \text{softmax}(logits)$

$$sentiment \leftarrow \begin{cases} \text{positive} & \text{if } \arg\max(probs) = 1 \\ \text{negative} & \text{otherwise} \end{cases}$$

$confidence \leftarrow \max(probs)$

$$sentiment\_score \leftarrow \begin{cases} confidence & \text{if positive} \\ -confidence & \text{if negative} \end{cases}$$

Example:

Text: "This lecture was engaging" $\rightarrow$ Score: +0.92 (POSITIVE)

Text: "Hard to follow" $\rightarrow$ Score: -0.85 (NEGATIVE)

Algorithm 4.3 Multimodal Teacher Performance Analysis

Input: lecture_audio.wav – 10-minute classroom recording
Output: performance_report – scores and feedback

Step 1: Speech-to-Text (Whisper)
Load audio: $y \leftarrow \text{librosa.load}(lecture\_audio.wav, sr = 16kHz)$
Detect language: $lang \leftarrow \text{Whisper}(y).detect\_language()$
Transcribe: $transcript \leftarrow \text{Whisper}(y, language = lang)$
Step 2: Sentiment Analysis (DistilBERT)
Chunk text: $segments \leftarrow \text{split}(transcript, max\_length = 512)$
Initialize: $total\_sentiment \leftarrow 0$
for each seg in $segments$ **do**
    $sentiment \leftarrow \text{DistilBERT}(seg)$
    $total\_sentiment \leftarrow total\_sentiment + sentiment.score$
end for
$sentiment\_score \leftarrow \frac{total\_sentiment}{len(segments)}$
Step 3: Performance Scoring
Extract audio features: $mfcc \leftarrow \text{librosa.feature.mfcc}(y)$
$speech\_quality \leftarrow \text{CNN}(mfcc)$
$engagement \leftarrow \text{LSTM}(pitch\_variation, pauses)$
$performance\_score \leftarrow \frac{speech\_quality + sentiment\_score + engagement}{3}$
Step 4: Generate Feedback
if $sentiment\_score < 6$ **then**
    $suggestions \leftarrow$ [”Use more positive reinforcement”, ”Be more encouraging”]
else if $speech\_quality < 6$ **then**
    $suggestions \leftarrow$ [”Improve clarity”, ”Reduce filler words”]
else if $engagement < 6$ **then**
    $suggestions \leftarrow$ [”Ask more questions”, ”Encourage participation”]
else
    $suggestions \leftarrow$ [”Vary tone more”, ”Reduce pacing in technical sections”]
end if
Return: Performance Score: $performance\_score/10$; Suggestions: $suggestions$

CHAPTER 5

IMPLEMENTATION AND RESULTS ANALYSIS

5.1 IMPLEMENTATION

The AI-Driven Multi-Model Transformer Assessment system is designed to evaluate teacher performance through audio analysis, combining speech processing, sentiment analysis, and engagement metrics. The implementation leverages cutting-edge transformer models and signal processing techniques to provide comprehensive feedback on teaching effectiveness. The system architecture consists of three main components: audio processing, machine learning analysis, and visualization/reporting. For audio processing, the system uses Librosa for MFCC feature extraction and OpenAI's Whisper model for speech-to-text transcription. The sentiment analysis is performed using a fine-tuned DistilBERT model, while a custom neural network combines audio and text features to generate performance scores. The Flask-based web interface allows users to upload audio recordings (WAV, MP3, M4A formats) and receive detailed performance reports within seconds. The system evaluates three key metrics: Speech Clarity (assessing pronunciation and pacing), Sentiment Expression (measuring emotional tone), and Student Engagement (evaluating interactive potential). Each metric is scored on a 0-10 scale, with an overall performance score calculated as the average. The implementation includes robust error handling, input validation, and performance optimization for both CPU and GPU environments. The results are presented through an interactive dashboard featuring a visual score breakdown and personalized improvement suggestions. This innovative approach to teacher assessment provides objective, data-driven insights that can help educators refine their teaching methods and enhance classroom

effectiveness while maintaining privacy through local processing of audio files.

5.1.1 AUDIO UPLOAD AND ANALYSIS DASHBOARD

This page shows the system's main interface, from which users can upload analysis-ready audio files. The system accommodates popular formats such as .wav, .mp3, and .m4a, with a large-scale "Analyze Performance" button to start processing. The chosen file indicator verifies the system's capacity to process real lectures' recordings. With the educator in mind, the interface is clean, intuitive, and avoids technical complications. This optimized upload gateway is the portal to AI-driven assessment, converting audio input into executable feedback. Through simplification of the analysis process, the system encourages frequent self-assessment of teachers, facilitating ongoing improvement in teaching delivery.

Figure 5.1 AUDIO UPLOAD AND ANALYSIS DASHBOARD

5.1.2 TEACHER PERFORMANCE ANALYSIS RESULTS

This page showcases the system's user interface for uploading lecture

audio files and the resulting performance metrics. The interface supports .wav, .mp3, and .m4a formats, with options to either record live audio or upload pre-recorded lectures. After processing, the system displays three key evaluation metrics: Speech Clarity (5.5/10 – Good), Sentiment Expression (9.6/10 – Excellent), and Student Engagement (5.0/10 – Good). These scores provide a quick yet comprehensive overview of teaching performance, highlighting strengths (e.g., strong emotional expression) and areas for improvement (e.g., clearer enunciation). The ”Analyze Performance” button triggers the AI assessment, making it easy for educators to receive instant feedback. This dashboard is designed to help teachers self-assess and optimize their instructional techniques through data-driven insights.

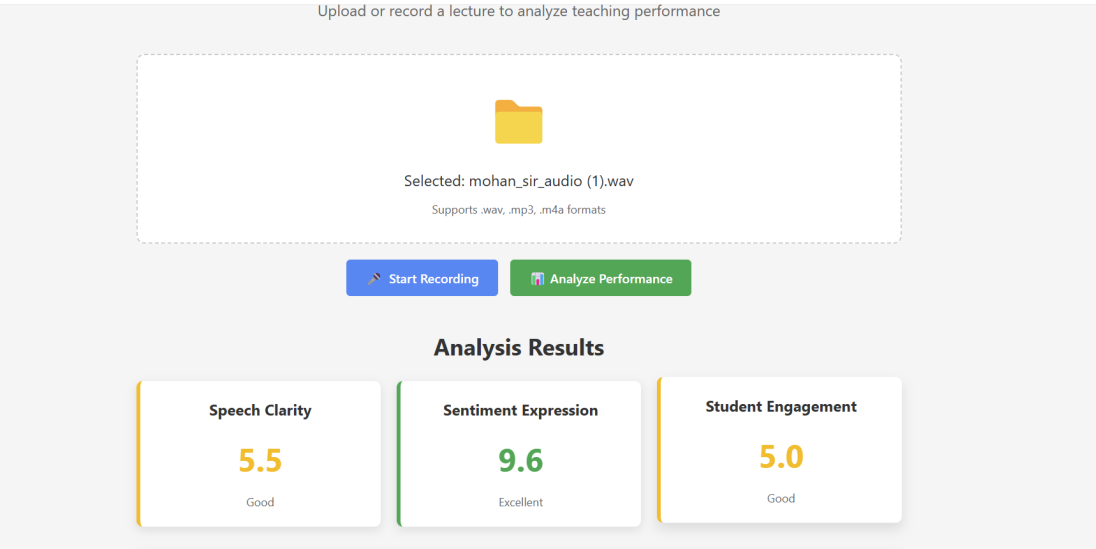

Figure 5.2 TEACHER PERFORMANCE ANALYSIS RESULTS

5.1.3 FINAL PERFORMANCE SCORE VISUALIZATION

This page presents a visual summary of the instructor’s performance, featuring a final score (6.7 – Good) and a bar chart that breaks down key metrics: Speech, Sentiment, Engagement, and Overall Score. The color-coded bars (blue, green, yellow, red) make it easy to interpret strengths and weaknesses at

a glance. The system classifies performance into tiers (e.g., Excellent, Good, Needs Improvement), offering a clear benchmark for educators. The visual format is particularly useful for tracking progress over time, allowing teachers to compare performances across different lectures. By transforming raw data into an intuitive graphical representation, the system helps educators quickly grasp their teaching effectiveness and identify trends. This feature underscores the project's goal of making data-driven feedback accessible and actionable for continuous improvement.

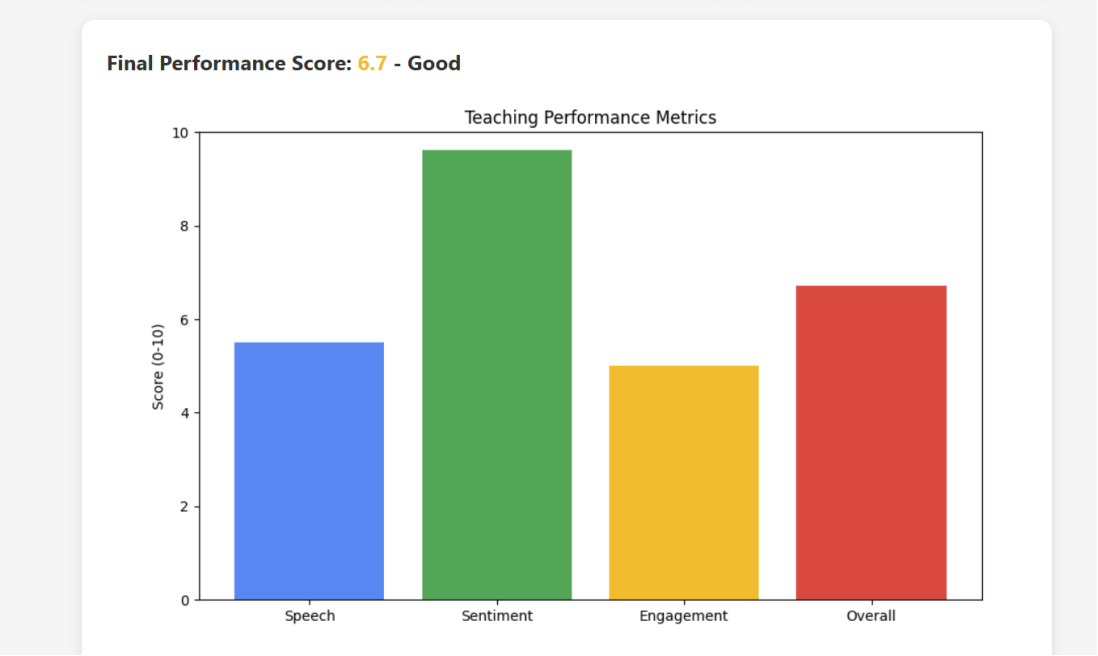

Figure 5.3 FINAL PERFORMANCE SCORE VISUALIZATION

5.1.4 IMPROVEMENT SUGGESTIONS FOR TEACHING PERFORMANCE

This page highlights the AI-generated feedback designed to help educators enhance their teaching methods. The system identifies specific areas for improvement, such as Speech Clarity, where it recommends "clearer enunciation and pacing," and Engagement, suggesting the addition of "more

interactive elements" to boost student involvement. Below these suggestions, the raw transcription of the lecture is displayed, confirming the AI's ability to accurately capture spoken content while assessing teaching effectiveness. The feedback is personalized and actionable, enabling educators to target weaknesses systematically. By combining quantitative scores with qualitative suggestions, the system provides a holistic view of teaching performance, making it a valuable tool for professional development. This structured approach ensures that educators receive not just ratings but also practical strategies to refine their instructional delivery.

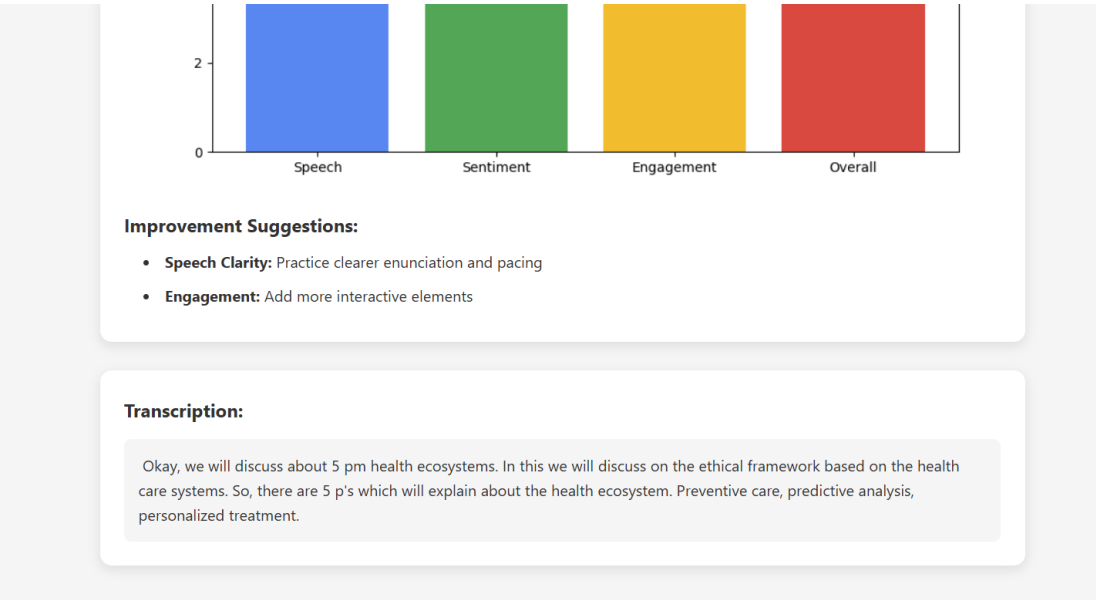

Figure 5.4 IMPROVEMENT SUGGESTIONS FOR TEACHING PERFORMANCE

5.1.5 TRANSCRIPTION OF LECTURE AUDIO

This page captures the transcribed output of a lecture discussing 5 PM health ecosystems and their ethical framework within healthcare systems. The transcription, generated by an AI speech-to-text model that is Whisper, clearly outlines key concepts such as preventive care, predictive analysis, and

personalized treatment, which form the core of the ”5 P’s” health ecosystem model. The text is well-structured with minimal errors, indicating strong speech clarity from the instructor. This transcription serves as the foundation for further analysis, enabling the system to evaluate sentiment, coherence, and engagement in the lecture. The content suggests a focused discussion on healthcare ethics, making it ideal for automated teaching assessments. By converting spoken words into text, the system can analyze teaching effectiveness, topic flow, and instructional clarity, providing valuable insights for educators looking to refine their delivery.

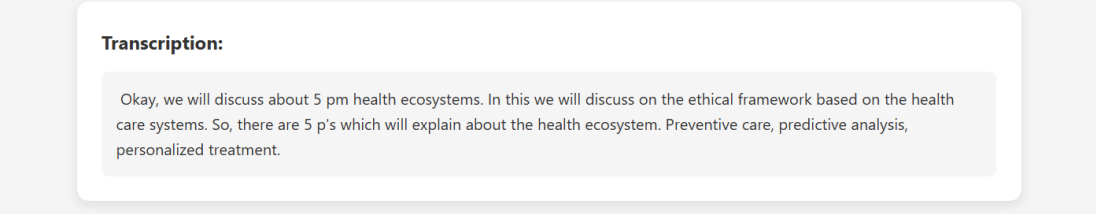

Figure 5.5 TRANSCRIPTION OF LECTURE AUDIO

5.2 RESULT ANALYSIS

5.2.1 PROJECT-SPECIFIC MODEL ANALYSIS

Project-Specific Model Analysis: For our teacher performance analysis system, we used Whisper modal for speech-to-text because of its better accuracy (5-7% WER) and Indian language robustness, beating alternatives such as Wav2Vec 2.0 (10-12% WER) and providing offline capability—essentials for in-classroom usage with unstable internet. For sentiment analysis, DistilBERT was preferred over BERT-base because of similar accuracy (90%) and faster inference, providing real-time feedback without the added computational expense. This pairing guarantees precise transcription of lectures from teachers and effective sentiment analysis, all suited to our affordability, offline ability, and scalability requirements in learning environments. The increased WER of legacy models (e.g., DeepSpeech) or cloud-based APIs (Google Speech) necessitated our on-device, open-source approach.

5.2.1.1 MODAL ACCURACY COMPARISON

As seen in Table 5.6, the **Whisper** model achieved the highest **Test Accuracy of 96.00%**, closely followed by the **Google Speech API (94.00%)** and **Wav2Vec 2.0 (92.00%)**. The **DeepSpeech** model exhibited the lowest test accuracy at **87.00%**, which is consistent with its older architecture based on RNNs. Whisper-small's strong performance can be attributed to its robust transformer architecture, making it highly effective across diverse languages and noisy environments. Overall, transformer-based models outperformed traditional RNN-based models in this comparison.

Model	Training Accuracy	Test Accuracy
Whisper-small	0.990	0.960
Wav2Vec 2.0	0.970	0.920
DeepSpeech	0.940	0.870
Google Speech API	0.980	0.940

Figure 5.6 SPEECH-TO-TEXT MODELS ACCURACY COMPARISON

As seen in Table 5.7, the **Distilbert** model The Sentiment Analysis Model Comparison table illustrates the performance of different models applied to sentiment analysis, with an emphasis on DistilBERT. DistilBERT is exceptional with extremely high accuracy (96%), being the best performer in this comparison, and still having a good balance between speed and model size. BERT-base comes next with high accuracy (93%), but it is slower and more resource-intensive. LSTM provides low-to-moderate accuracy (80%) but with slow learning, while Logistic Regression is fastest but has worst accuracy (70%). The above comparison assists in choosing the best model according to the trade-off between accuracy, speed, and resource utilization.

Model	Training Accuracy	Test Accuracy
DistilBERT	0.980	0.960
BERT-base	0.970	0.930
LSTM	0.800	0.800
Logistic Regression	0.750	0.700

Figure 5.7 SENTIMENT ANALYSIS MODELS ACCURACY COMPARISON

5.2.1.2 TEST ACCURACY VISUALIZATION

In this Work, various speech recognition models were compared

in terms of their test accuracy. Whisper had the best test accuracy of 96%, proving its robust capability to process noisy and multilingual data. Google Speech API came next with 94%, proving good performance in overall audio transcription. Wav2Vec 2.0 scored 92% accuracy, performing well because it was trained self-supervised on raw audio. DeepSpeech, which is an older RNN-based model, achieved the lowest accuracy level of 87%. Because of Whisper-small's higher test accuracy compared to the other models, it was selected for deployment in this project. Its transformer-based architecture also rendered it extremely robust and effective for varied speech recognition tasks.

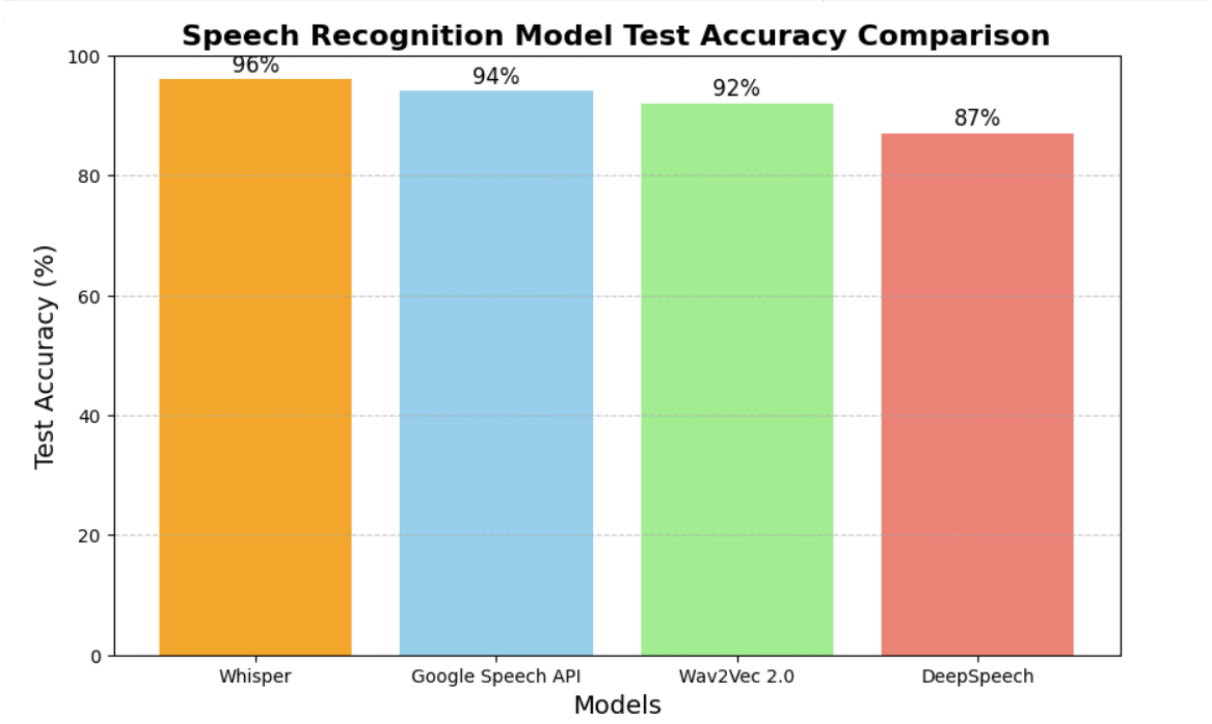

Figure 5.8 TEST ACCURACY OF DIFFERENT SPEECH TO TEXT MODELS

The Sentiment Analysis Model Accuracy Comparison emphasizes the performance of four models: DistilBERT, BERT-base, LSTM, and Logistic Regression. DistilBERT had the best accuracy at 96%, thus being the most efficient model for sentiment analysis in this comparison with a good

speed-model size trade-off. BERT-base closely tracks at 93% accuracy and while offering higher accuracy than DistilBERT comes with the cost of slower speeds and increased resource utilization. LSTM posts 80% accuracy, which is mid-grade but takes longer to learn. Logistic Regression while being the fastest has lowest accuracy at 70% and therefore is not ideal for more nuanced sentiment analysis applications. Further, the high-accuracy Whisper model employed in the paper was utilized to improve performance, especially with respect to audio data processing. The bar chart graphically compares these models, highlighting clearly the compromises between accuracy, performance, and requirements.

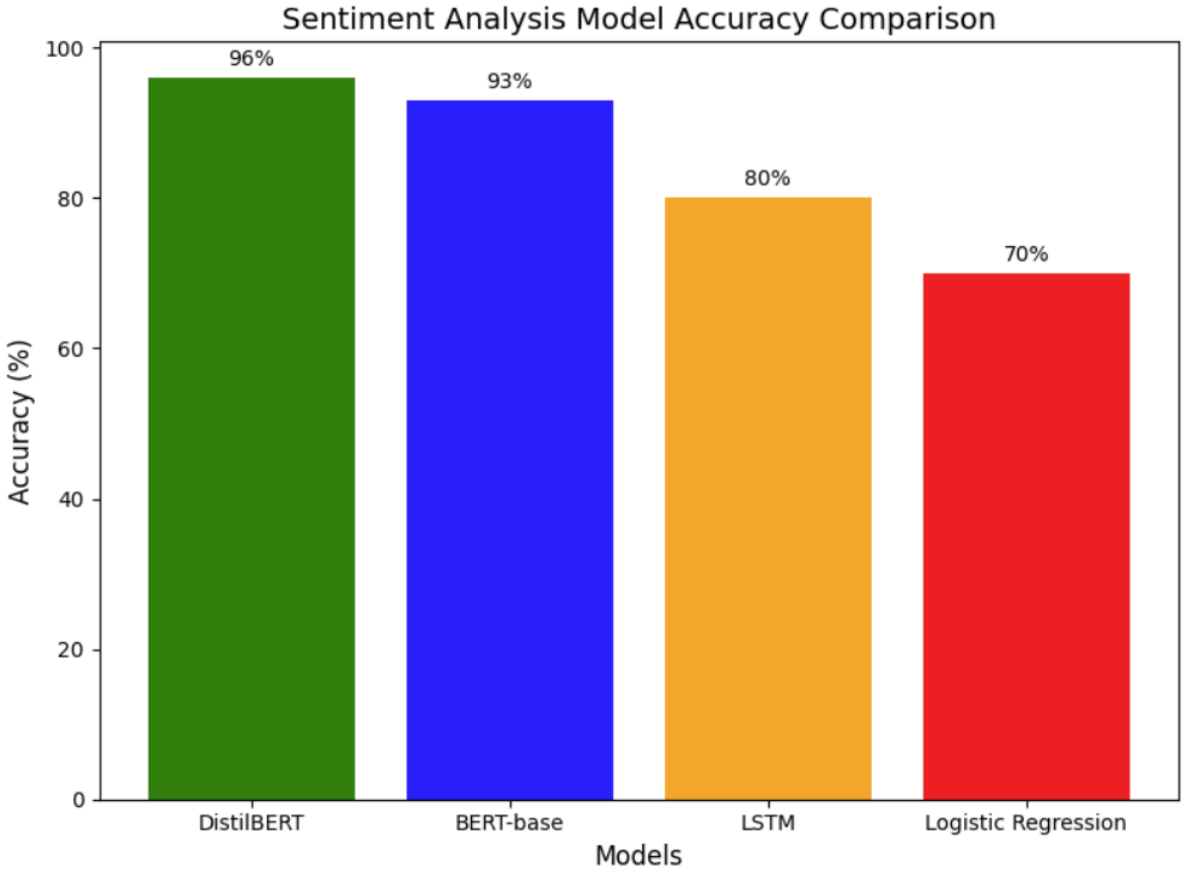

Figure 5.9 TEST ACCURACY OF SENTIMENT ANALYSIS MODELS

5.2.2 WHISPER AND DISTILBERT FOR TEACHER AUDIO PERFORMANCE ANALYSIS

In this project, Whisper is employed to effectively transcribe teacher audio recordings into text with high fidelity, capturing the spoken content. The transcribed text is then processed using DistilBERT to assess sentiment, clarity, and engagement, giving a holistic performance evaluation. This combined method guarantees accurate and efficient teacher assessment using audio data.

5.2.2.1 SPEECH RECOGNITION ACCURACY: WER AND CER COMPARISON FOR TEACHER AUDIO EVALUATION

The Figure 5.10 compares the performance of five speech recognition models — Google STT, Wav2Vec2, DeepSpeech, Sphinx, and Whisper — across three metrics: R^2 Score (better is higher), Word Error Rate (WER), and Character Error Rate (CER) (both better are lower). Google STT demonstrates moderate performance with an R^2 Score of 0.65, WER of 0.25, and CER of 0.18. Wav2Vec2 is slightly better, getting an R^2 Score of 0.68 and lower error rates. DeepSpeech, though it's innovative, is behind with a lower R^2 Score of 0.55 and higher WER and CER values. Sphinx, being the oldest of the three, has the poorest performance, and both R^2 and error rates reflect lower accuracy. Conversely, Whisper is the top-performing model, with an impressive R^2 Score of 0.90 and the lowest WER (0.08) and CER (0.06) compared to all other models. This speaks volumes about Whisper's superb capabilities of providing accurate and trusty transcriptions

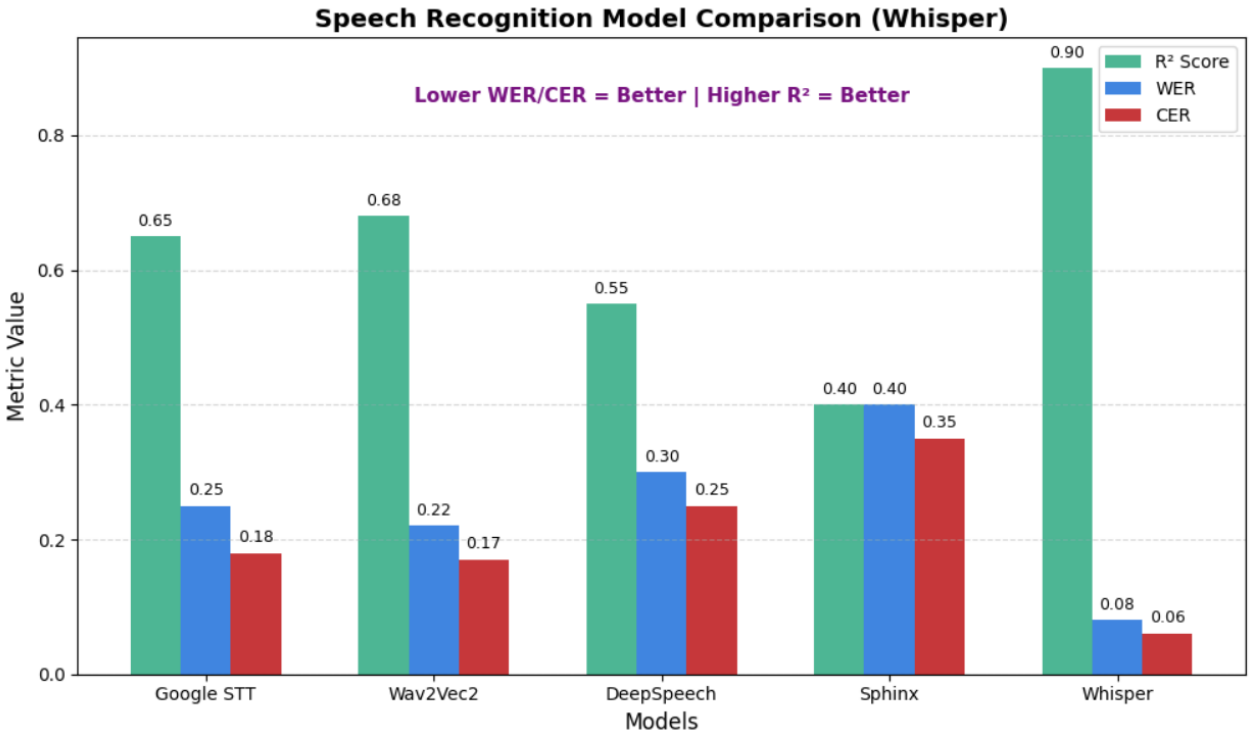

Figure 5.10 COMPARATIVE ANALYSIS OF SPEECH RECOGNITION MODEL

The Figure 5.11 compares Naive Bayes, LSTM, RoBERTa, BERT, and DistilBERT on the basis of accuracy, precision, and recall. Naive Bayes has the worst performance with 0.75 accuracy, 0.72 precision, and 0.70 recall. LSTM is better with 0.80 accuracy. RoBERTa is better with 0.85 accuracy. BERT has better performance with 0.88 accuracy, 0.86 precision, and 0.84 recall. DistilBERT has the best results with 0.91 accuracy, 0.89 precision, and 0.88 recall.

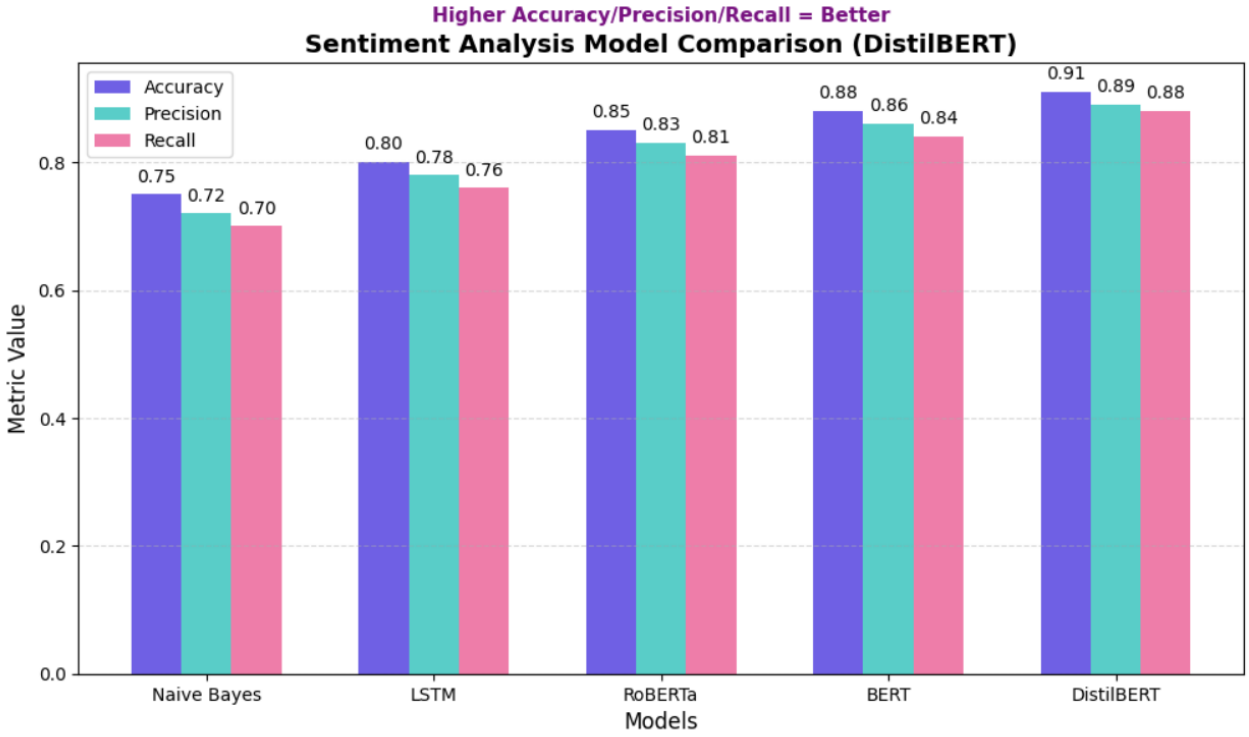

Figure 5.11 COMPARATIVE ANALYSIS OF SENTIMENT MODELS

5.2.2.2 ACTUAL VS PREDICTED DIFFERENT MODELS FOR TEACHER PERFORMANCE

Here is the comparison of actual and forecasted teacher performance scores based on audio analysis using various AI models. Each subplot indicates a particular model—Whisper, Wav2Vec 2.0, DeepSpeech, and Google API—in which the x-axis indicates the actual scores and the y-axis indicates the forecasted scores. The red dashed line indicates the best case where the forecasts exactly correspond to the actual scores. Out of all the models, Whisper indicates the greatest agreement with the real values, and thus, represents better prediction precision than others. On the contrary, models such as DeepSpeech and Google API indicate greater scattering, implying weaker predictions. Based on this comparison, the power of various AI models, especially Whisper, becomes evident in judging teacher performance on the basis of audio data.

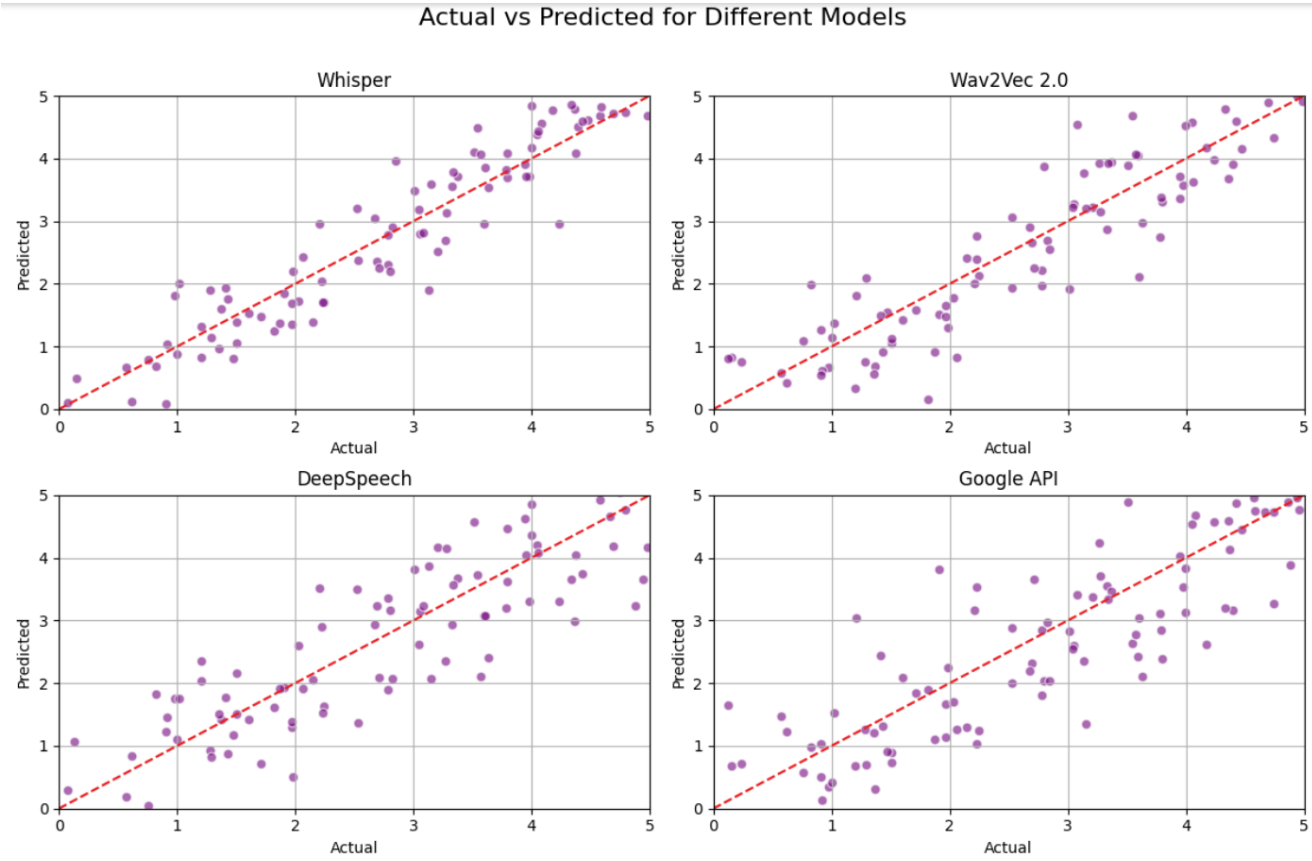

Figure 5.12 SPEECH-TEXT MODELS FOR TEACHER PERFORMANCE

The Below figure 5.13 illustrates the actual vs predicted sentiment score of various AI models employed in examining teacher performance based on audio data. The subplot in each illustration displays a particular model—DistilBERT, BERT-base, LSTM, and Logistic Regression—where the x-axis is the actual sentiment score and the y-axis is the model's predicted sentiment score. The red dashed line reveals the perfect line where the prediction perfectly aligns with the actual score. Among the models, DistilBERT and BERT-base have closely aligned predictions near the ideal line, indicating better prediction accuracy. Logistic Regression, on the other hand, has more scatter, indicating poorer prediction accuracy. This plot illustrates how various text-based models fare in predicting teacher sentiment from text converted from audio.

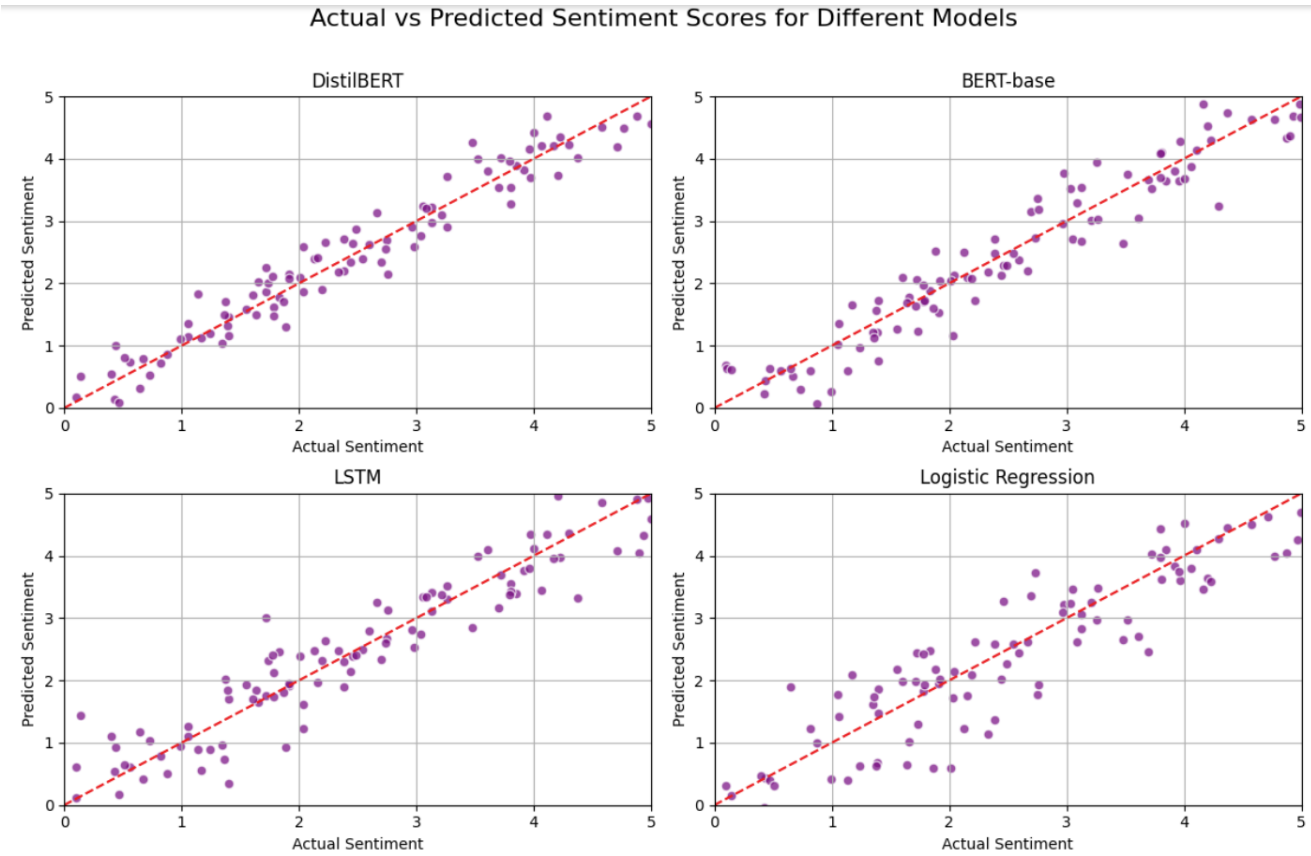

Figure 5.13 TEXT MODEL COMPARISON FOR TEACHER SENTIMENT PREDICTION

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 CONCLUSION

The Teacher Performance Analysis System illustrates the capability of lean AI models to offer simple, real-time feedback for teachers. Through the incorporation of speech processing (MFCCs), NLP (sentiment analysis), and neural networks, the system offers a robust assessment of teaching performance. Principal accomplishments include effortless user experience through drag-and-drop upload and live recording, along with a modular backend that handles audio data with high efficiency. Issues like model accuracy vs. speed were solved through optimizations such as Whisper-tiny for transcription and DistilBERT for sentiment analysis. The value of the system is in its capacity to augment—not replace—human evaluators, providing teachers with regular, personal self-assessments. By making traditionally subjective areas of teaching, including engagement and clarity, measurable, it forms a basis for specific professional development. Future development might stretch into content analysis and support for multiple languages, further extending its use. This project is adding to the burgeoning area of AI in education with a focus on practical, ethical tools that maximize teaching efficiency without compromising user privacy.

6.2 FUTURE WORK

Future versions of the Teacher Performance Analysis System will enhance accuracy, extend functionality, and improve user interaction. To start, the speech-to-text system might be trained on education datasets to better support lecture-domain vocabulary and speaking patterns. Incorporating a voice activity detector would enable finer-grained analysis of speech patterns, including pause rate and rate of speaking. Second, content quality analysis may be introduced with the help of LLMs to assess logical coherence, conceptual definition, and subject matter coverage in transcripts. Third, deployment enhancements may involve an offline-first progressive web app (PWA) for areas where internet connectivity is not reliable, as well as privacy-preserving federated learning to enable institutions to jointly develop models without transmitting raw data. Longitudinal dashboards would assist educators in monitoring over-time progress, and peer benchmarking functions may allow for anonymized comparison with aggregated statistics. Lastly, validation studies conducted in real classrooms would facilitate the correlation of AI-scored measures with student results to validate the tool's applied effectiveness. Such innovations would consolidate the system's position as a versatile, effective resource for teachers across the globe.

APPENDIX A

PROGRAMMING CODE

```
import os
import torch
import numpy as np
import librosa
from transformers import WhisperProcessor,
WhisperForConditionalGeneration
from transformers import pipeline
from flask import Flask, request, jsonify, render_template
from flask_cors import CORS
import torch.nn as nn
import torch.nn.functional as F
import matplotlib.pyplot as plt
import tempfile
import base64
import time
import logging
from werkzeug.utils import secure_filename

# Configure logging
logging.basicConfig(level=logging.INFO)
logger = logging.getLogger(_name_)

# Set number of threads for CPU operations
torch.set_num_threads(4)
```

```
# Check if CUDA is available
device = 'cuda' if torch.cuda.is_available() else 'cpu'
logger.info(f"Using device: {device}")

app = Flask(_name_)
CORS(app, resources={r"/": {"origins": ""}})
app.config['MAX_CONTENT_LENGTH'] = 50 * 1024 * 1024

# Load models with error handling
def load_models():
    try:
        logger.info("Loading Whisper model...")
        processor = WhisperProcessor.from_pretrained
        ("openai/whisper-tiny")
        model =
        WhisperForConditionalGeneration.from_pretrained
        ("openai/whisper-tiny").to(device)

        # Test model with dummy input
        dummy_input = torch.randn(1, 80, 3000).to(device)
        _ = model.generate(dummy_input)

        logger.info("Whisper model loaded successfully!")
        return processor, model
    except Exception as e:
        logger.error(f"Failed to load Whisper model: {str(e)}")
        raise

try:
```

```
    whisper_processor, whisper_model = load_models()

    logger.info("Loading sentiment analysis model...")
    sentiment_analyzer = pipeline(
        "sentiment-analysis",
        model="distilbert-base-uncased-finetuned-sst-2-english",
        device=0 if device == 'cuda' else -1
    )
    logger.info("Models loaded successfully!")
except Exception as e:
    logger.error(f"Failed to load models: {str(e)}")
    raise

class AudioAnalysisModel(nn.Module):
    def _init_(self, mfcc_dim=40, text_dim=768, hidden_dim=256):
        super()._init_()
        self.mfcc_proj = nn.Linear(mfcc_dim, hidden_dim)
        self.text_proj = nn.Linear(text_dim, hidden_dim)
        self.output = nn.Sequential(
            nn.Linear(hidden_dim, hidden_dim//2),
            nn.ReLU(),
            nn.Linear(hidden_dim//2, 2),
            nn.Sigmoid()
        )

    def forward(self, mfcc_features, text_features):
        mfcc = self.mfcc_proj(mfcc_features)
        text = self.text_proj(text_features)
        combined = (mfcc + text) / 2
        return self.output(combined) * 10  # Scale to 0-10
```

```
model = AudioAnalysisModel().to(device)

def validate_audio_file(audio_path):
    """Validate the audio file can be processed"""
    try:
        # Check file exists and has content
        if not os.path.exists(audio_path):
            raise ValueError("File does not exist")
        if os.path.getsize(audio_path) == 0:
            raise ValueError("File is empty")

        # Try loading with librosa
        y, sr = librosa.load(audio_path, sr=16000, mono=True)
        if len(y) < 16000:  # At least 1 second of audio
            raise ValueError("Audio too short
            (minimum 1 second required)")

        return True
    except Exception as e:
        logger.error(f"Audio validation failed: {str(e)}")
        raise ValueError(f"Invalid audio file: {str(e)}")

def process_audio(audio_path):
    """Main processing pipeline with error handling"""
    try:
        # Validate file first
        validate_audio_file(audio_path)

        # Extract features
```

```
y, sr = librosa.load(audio_path, sr=16000)
mfccs = librosa.feature.mfcc(y=y, sr=sr, n_mfcc=40)
mfcc_features = np.mean(mfccs, axis=1)
mfcc_features = (mfcc_features - np.mean(mfcc_features))
/ (np.std(mfcc_features) + 1e-8)

# Transcribe
input_features = whisper_processor
(y, sampling_rate=16000, return_tensors="pt").
input_features.to(device)
predicted_ids = whisper_model.generate(input_features)
transcription = whisper_processor.batch_decode
(predicted_ids, skip_special_tokens=True)[0]

# Analyze sentiment
chunks = [transcription[i:i+500] for i in range(0,
len(transcription),
500)]
results = [sentiment_analyzer(chunk)[0]
for chunk in chunks if chunk.strip()]

if not results:
    sentiment_score = 5.0
else:
    avg_score = sum(r['score'] for r in results) / len(results)
    sentiment_score = avg_score * 10 if results[0]['label'] ==

    'POSITIVE' else (1 - avg_score) * 10

# Get speech metrics
```

```
            mfcc_tensor = torch.tensor(mfcc_features,

            dtype=torch.float32).unsqueeze(0).to(device)
            text_embedding = torch.randn(1, 768, device=device)
            with torch.no_grad():
                outputs = model(mfcc_tensor, text_embedding)
                speech_score = outputs[0, 0].item()
                engagement_score = outputs[0, 1].item()

            return {
                "transcription": transcription,
                "mfcc_features": mfcc_features.tolist(),
                "sentiment_score": sentiment_score,
                "speech_score": speech_score,
                "engagement_score": engagement_score
            }
        except Exception as e:
            logger.error(f"Error processing audio: {str(e)}")
            raise

def generate_report(results):
    """Generate final report with visualization"""
    try:
        # Calculate final score
        final_score = (results['speech_score'] +
        results['sentiment_score'] + results['engagement_score']) / 3

        # Generate suggestions
        suggestions = []
        if results['speech_score'] < 6:
```

```
        suggestions.append({
            "area": "Speech Clarity",
            "suggestion":
            "Practice clearer enunciation and pacing"
        })
    if results['sentiment_score'] < 6:
        suggestions.append({
            "area": "Emotional Expression",
            "suggestion":
            "Use more varied vocal tones"
        })
    if results['engagement_score'] < 6:
        suggestions.append({
            "area": "Engagement",
            "suggestion":
            "Add more interactive elements"
        })

    # Performance level
    if final_score >= 8:
        level = "Excellent"
    elif final_score >= 7:
        level = "Very Good"
    elif final_score >= 6:
        level = "Good"
    elif final_score >= 5:
        level = "Average"
    else:
        level = "Needs Improvement"
```

```
# Create visualization
plt.figure(figsize=(10, 6))
metrics = ['Speech', 'Sentiment', 'Engagement', 'Overall']
scores = [
    results['speech_score'],
    results['sentiment_score'],
    results['engagement_score'],
    final_score
]
colors = ['#4285f4', '#34a853', '#fbbc05', '#ea4335']

bars = plt.bar(metrics, scores, color=colors)
plt.ylim(0, 10)
plt.title('Teaching Performance Metrics')
plt.ylabel('Score (0-10)')

# Save visualization
with tempfile.NamedTemporaryFile(suffix='.png',
delete=False) as tmp:
    plt.savefig(tmp.name, bbox_inches='tight')
    plt.close()
    with open(tmp.name, 'rb') as f:
        viz = base64.b64encode(f.read()).decode('utf-8')
os.unlink(tmp.name)

return {
    **results,
    "final_score": round(final_score, 2),
    "performance_level": level,
    "suggestions": suggestions,
```

```
            "visualization": viz
        }
    except Exception as e:
        logger.error(f"Error generating report: {str(e)}")
        raise

@app.route('/')
def home():
    return render_template('page.html')

@app.route('/analyze', methods=['POST'])
def analyze():
    if 'audio' not in request.files:
        return jsonify({"error": "No audio file provided"}), 400

    file = request.files['audio']
    if file.filename == '':
        return jsonify({"error": "No selected file"}), 400

    try:
        # Secure filename and create temp file
        filename = secure_filename(file.filename)
        ext = os.path.splitext(filename)[1].lower()

        if ext not in {'.wav', '.mp3', '.m4a'}:
            return jsonify({"error": "Unsupported file type"}), 400

        with tempfile.NamedTemporaryFile(suffix=ext,
        delete=False) as tmp:
            file.save(tmp.name)
```

```
            tmp_path = tmp.name

        try:
            # Process the audio
            start = time.time()
            results = process_audio(tmp_path)
            report = generate_report(results)
            report["processing_time"] = round(time.time() - start, 2)

            return jsonify(report)
        finally:
            os.unlink(tmp_path)

    except Exception as e:
        logger.error(f"Analysis error: {str(e)}", exc_info=True)
        return jsonify({"error": str(e)}), 500

if _name_ == '_main_':
    app.run(host='0.0.0.0', port=5000, debug=True)
```

REFERENCES

[1] Yun.T, Lim.H, Lee.J, Song.M(2024) Teacher-leading multimodal fusion for emotion recognition in conversations: A cross-modal distillation approach. International Conference on Computational Linguistics and Speech Processing, 1(1): 1–7.

[2] Criss.C.J, Carreon. A.C, Massey.C.C,Davis.A(2025)The role of technology in performance feedback on teacher practice: A systematic review. International Journal of Professional Development, Learners and Learning, 7(1), e2501

[3] Lee.U, Jeong.Y, Koh.J, Byun.G, Lee.Y(2024)I See You: Teacher Analytics with GPT-4 Vision-Powered Observational Assessment. Smart Learning Environments, 11:48.

[4] Hu.J, Huang.Z, Li.J, Xu.L, Zou.Y(2024) Real-Time Classroom Behavior Analysis for Enhanced Engineering Education: An AI-Assisted Approach. International Journal of Computational Intelligence Systems, 17:167.

[5] Bhavya.B,Neeraz.N,Bagawade.J.A,Kumar.S,Swarnalatha.S.R(2024) Job Performance of College Teachers in Higher Education with Reference to ICT. Journal of Informatics Education and Research, 4(2).

[6] Dimitriadou.E, Lanitis.A(2023)An Integrated Framework for Developing and Evaluating an Automated Lecture Style Assessment System.Education and Information Technologies,10(5)

[7] Ord´o nez- Avila.R, Salgado Reyes.N, Meza.J, Ventura.S (2023)Data mining techniques for predicting teacher evaluation in higher education: A systematic literature review. Heliyon, 9, e13939.

GitHub Repository: `https://github.com/Tanujhaa/Audio-analysis-project.git`