Machine Learning Approach to Sentiment Analysis in Telephonic Conversation

By

Qasim Mehmood

22-ARID-707

Syed Rafay Ali

22-ARID-728

Moiz Abdullah

22-ARID-656

Supervisor

Hafiz Muhammad Faisal

University Institute of Information Technology,

PMAS-Arid Agriculture University,

Rawalpindi Pakistan

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction

This chapter provides an overview of the project, including its background, relevance, literature review, and methodology. The project aims to develop an AI-powered multimodal system for sentiment analysis in telephonic conversations, combining Natural Language Processing and Speech Emotion Recognition to provide comprehensive insights into customer-agent interactions.

It introduces the core concept of multimodal sentiment analysis in telephonic conversations, discusses the motivation behind developing a hybrid deep learning framework integrating speech emotion recognition with natural language processing, and establishes the software development lifecycle and methodology adopted throughout this research.

1.1. Brief

This project presents a machine learning–based system designed for **sentiment analysis in telephonic conversations**, aimed at understanding not only what customers say but also how they say it. Traditional sentiment analysis methods often rely solely on text data, overlooking the valuable emotional and behavioral signals carried in speech. To overcome this limitation, the proposed system combines **Automatic Speech Recognition (ASR)**, **Natural Language Processing (NLP)**, and **Speech Emotion Recognition (SER)** to analyze both textual and acoustic aspects of customer–agent interactions.

The system uses **Whisper ASR** for highly accurate speech-to-text transcription and **Pyannote.audio** for speaker diarization to separate voices in multi-speaker conversations. For textual sentiment classification, **DistilBERT** and **FinBERT** transformer models are employed, while a **CNN+LSTM** hybrid network captures emotional cues from tone, pitch, and speech dynamics. The extracted features are then fused using **XGBoost** for predictive modeling, enabling the system to forecast **sales conversion probabilities** and overall customer satisfaction.

Developed through **Agile methodology** with iterative testing and improvement, the final solution provides **interactive dashboards** that display real-time sentiment trends, emotion tracking, and performance insights offering organizations a powerful and cost-effective tool to enhance customer experience and decision-making.

Unlike existing commercial solutions that provide partial multimodal analysis at prohibitive costs, this project delivers an open-source, academically rigorous, and deployment-ready framework specifically designed for call center sentiment analysis, sales conversion prediction, and agent performance evaluation. The system outputs include real-time sentiment trajectories, emotion classification, conversational dynamics modeling (interruptions, turn-taking, sentiment drift), and probabilistic sales conversion forecasts presented through interactive web-based dashboards.

1.2. Relevance to Course Modules

This project is directly linked to multiple subjects studied during the BS Artificial Intelligence degree, including:

Course	Relevance
Artificial Intelligence / Machine Learning	Core AI/ML concepts such as supervised learning, clustering (k-means), optimization, loss functions, and evaluation metrics (DER, JER) are used to train and evaluate the EEND-NAA model.
Deep Learning	EEND-NAA architecture is purely deep-learning based — it uses Transformer encoders and decoders, multi-head attention, embedding generation and non-autoregressive attractor refinement.
Natural Language Processing (NLP)	Sequence modeling and Transformer architectures.
Software Engineering	Linguistic modeling using ASR outputs and text embeddings.
Speech Processing	Acoustic feature extraction (MFCCs, pitch, energy, zero-crossing rate), speech-to-text transcription (Whisper ASR), speaker diarization, prosody analysis, and audio signal preprocessing (noise reduction, normalization).
Web Technologies	Development of interactive dashboard using React and Flask/Fast API

1.3. Project Background

Customer-agent interactions in call centers hold valuable insights into customer mood, intent, and buying probability. Traditional sentiment analysis focuses only on binary positive/negative classification, but real conversations involve multi-faceted features such as tone, pitch, speech pauses, conversational context, and emotional dynamics. Understanding these nuances is crucial for businesses to improve customer satisfaction, optimize agent performance, and increase sales conversion rates.

The telecommunications and customer service industries generate millions of call recordings daily, yet most organizations lack the tools to extract meaningful insights from this data. Existing solutions are often expensive, proprietary, and limited in their analytical capabilities. This project addresses these gaps by developing an open-source, cost-effective, and comprehensive solution that combines multiple AI technologies to provide deeper insights into customer-agent conversations.

The motivation for this project emerged from three converging factors:

1. **Academic Gap:** Limited research on production-ready multimodal sentiment analysis systems specifically designed for telephonic conversations, with most academic work focusing on acted emotional speech datasets rather than naturalistic call center interactions.
2. **Commercial Limitations:** Existing enterprise solutions (IBM Watson, Amazon Contact Lens, Google Cloud) are proprietary, expensive (typically $0.05-$0.15 per conversation minute), lack customization flexibility, and provide limited explainability of model predictions.
3. **Business Need:** Call center managers and sales analysts require cost-effective, transparent, and actionable sentiment analysis tools that integrate seamlessly with existing CRM systems and provide real-time performance monitoring.

This project bridges these gaps by developing an open-source, explainable, and empirically validated multimodal sentiment analysis framework that advances both academic research and practical call center applications.

1.4. Literature Review

Several commercial and research systems have been developed for conversation analysis, each with specific strengths and limitations:

IBM Watson Tone Analyzer: IBM's solution provides emotion detection and tone analysis from text data. It can identify emotions such as joy, fear, sadness, and anger, as well as language tones like analytical, confident, and tentative. However, it lacks support for prosodic features in speech such as pitch, tone, and acoustic patterns, limiting its effectiveness for telephonic conversation analysis (IBM, 2025).

Amazon Contact Lens: Amazon Web Services offers Contact Lens for Amazon Connect, which provides speech-to-text transcription and sentiment analysis for contact center conversations. While it includes some speech analysis capabilities, it is a proprietary solution with high costs, making it inaccessible for small to medium-sized businesses and academic research. Additionally, it does not provide sales forecasting or conversion probability prediction

Google Cloud Speech and Natural Language APIs: Google Cloud offers separate services for speech-to-text transcription and sentiment analysis. While these services are powerful individually, they require integration and lack built-in support for conversational dynamics modeling, emotion recognition from acoustic features, and predictive analytics for sales outcomes (Google Cloud, 2025).

1.4.1 Traditional Sentiment Analysis Approaches

Early sentiment analysis systems relied on lexicon-based methods employing manually curated dictionaries mapping words to sentiment scores. VADER (Valence Aware Dictionary and sEntiment Reasoner) and TextBlob represent widely adopted lexicon-based tools that analyze textual sentiment through rule-based pattern matching and polarity aggregation. While computationally efficient and interpretable, these approaches suffer from fundamental limitations:

- Context Insensitivity: Lexicon methods fail to capture contextual sentiment shifts, sarcasm, and domain-specific language nuances.
- Binary Classification: Reduction of complex emotional states to positive/negative/neutral categories ignores emotional intensity and multi-dimensional affect.
- Text-Only Scope: Complete disregard of acoustic prosodic features carrying critical emotional information in spoken communication.

Supervised machine learning approaches using Naive Bayes, Support Vector Machines (SVM), and Random Forests improved contextual understanding by learning patterns from labeled training data. However, these traditional ML methods require extensive manual feature engineering and struggle with high-dimensional feature spaces characteristic of conversational audio-text data.

1.4.2 Research Gap

Based on comprehensive literature review, the following critical gaps emerge:

1. Lack of Integrated Multimodal Systems: No open-source framework exists that seamlessly integrates state-of-the-art ASR, NLP, SER, and predictive analytics specifically designed for call center applications.
2. Limited Explainability: Most deep learning systems operate as black boxes, providing predictions without interpretable explanations—a significant barrier to call center adoption where managers require actionable insights and justifiable agent feedback.
3. Absence of Conversational Dynamics Modeling: Current systems analyze isolated utterances rather than modeling conversation-level phenomena (interruption patterns, sentiment drift, agent responsiveness) that strongly correlate with interaction outcomes.
4. Generalization Limitations: Models trained on acted emotional speech or product review datasets fail to generalize to authentic call center conversations characterized by domain-specific terminology, background noise, and naturalistic emotional expression.
5. Cost and Accessibility Barriers: Commercial systems remain prohibitively expensive for small-to-medium enterprises and academic research, while research prototypes lack deployment readiness.

This project directly addresses these gaps by developing an integrated, explainable, conversation-aware, and deployment-ready multimodal sentiment analysis system specifically optimized for telephonic customer-agent interactions.

1.5. Analysis from Literature Review

Critical analysis of existing literature reveals both methodological strengths to leverage and fundamental limitations to overcome in developing the proposed multimodal sentiment analysis system.

1.5.1 Comparative Analysis of Existing Approaches

Table 1 Comparison of Existing Approaches

Approach	Strengths	Limitations
Deep Learning NLP (BERT, DistilBERT)	Contextual embeddings capture semantic nuances; transfer learning from large corpora; handles long-range dependencies	Computationally intensive; requires large training datasets; still text-only; lacks acoustic awareness
Speech Emotion Recognition (CNN+LSTM)	Captures prosodic emotional cues; learns hierarchical acoustic representations; temporal modeling	Content-agnostic; poor generalization from acted speech datasets; discrete emotion categories; audio-only analysis
Lexicon-Based Sentiment Analysis (VADER, TextBlob)	Computationally efficient; interpretable rules; no training data required; domain-portable	Context-insensitive; fails on sarcasm/negation; ignores acoustic prosody; limited to discrete sentiment categories
Commercial Multimodal Systems (IBM Watson, Amazon Contact Lens)	Production-ready infrastructure; some multimodal capabilities; enterprise support	Proprietary black boxes; prohibitively expensive; limited customization; weak explainability; minimal predictive analytics

1.5.2 Key Observations

Analysis reveals several consistent patterns across existing literature:

1. Modality Complementarity: Studies consistently demonstrate that multimodal fusion (text + audio) outperforms unimodal approaches, with typical accuracy improvements of 8-15% over text-only baselines.
2. Architectural Superiority of Deep Learning: Transformer-based NLP models and CNN+LSTM acoustic models consistently outperform traditional machine learning approaches, particularly when trained on large-scale datasets.
3. Importance of Domain Adaptation: Models trained on general-purpose datasets (movie reviews, social media) exhibit significant performance degradation when applied to call center conversations, emphasizing the need for domain-specific training data and fine-tuning.

1.5.3 Explainability and Transparency:

The system incorporates attention weight visualization, SHAP (SHapley Additive exPlanations) value computation, and feature importance ranking to provide interpretable explanations for predictions—essential for call center manager trust and actionable agent feedback.

1.5.4Open-Source and Cost-Effective:

All components utilize open-source technologies (PyTorch, Hugging Face Transformers, Pyannote.audio), eliminating per-conversation licensing fees and enabling academic reproducibility and community extension

1.5.5 Summary

The proposed multimodal sentiment analysis system synthesizes best practices from NLP, speech processing, and predictive analytics while addressing critical gaps in existing research and commercial offerings. By integrating state-of-the-art ASR, transformer-based text analysis, deep learning acoustic emotion recognition, conversational dynamics modeling, and ensemble-based conversion prediction within an open-source framework, the system advances both academic research and practical call center applications. The explicit focus on explainability, domain-specific optimization, and business intelligence integration distinguishes this work from prior research and positions it for real-world deployment impact.

1.6. Methodology and Software Lifecycle for This Project

This section describes the comprehensive methodology and Software Development Life Cycle (SDLC) model adopted for developing the multimodal sentiment analysis system. Given the project's hybrid nature—combining research experimentation with production software engineering—a traditional linear development model proves insufficient. Therefore, the project employs an Agile-based Incremental Methodology emphasizing iterative development, continuous integration, and empirical validation.

Methodology Overview :

The methodology is based on a hybrid deep learning approach consisting of major modules:

Module	Function
Acoustic Processing Stream	Converts raw audio to acoustic features; performs speaker diarization; extracts prosodic features (pitch, energy, MFCCs); recognizes emotional states from speech signals
Linguistic Processing Stream	Transcribes speech to text; performs text preprocessing (tokenization, lemmatization); conducts sentiment classification; analyzes semantic content
Conversational Dynamics Stream	Models turn-taking patterns; detects interruptions and overlaps; quantifies hesitation cues; tracks sentiment trajectories; computes agent responsiveness metrics
Fusion and Prediction Module	Integrates multimodal features; performs attention-based feature fusion; predicts sales conversion probability; generates explainable insights
Presentation Layer	Provides interactive web dashboards; visualizes sentiment trajectories and emotion distributions; displays conversion probability metrics; generates analytical reports

This modular methodology enables parallel development by team members, facilitates independent testing of subsystems, and supports incremental integration with systematic validation at each stage.. This project employs the **Agile development methodology** with iterative sprints to ensure flexibility, continuous improvement, and stakeholder feedback integration. The Agile approach is particularly suitable for this project due to its experimental nature and the need for continuous model refinement and testing.

1.6.1. Software Development Lifecycle (SDLC) Model

The project adopts the Incremental Agile Model structured around two-week sprints with clearly defined deliverables:

Table 2 Development Phases

Phase	Description
Week 1-2	Understanding functional and non-functional requirements, dataset selection, system architecture planning.
Week 3-4	Dividing work into three modules and assigning responsibilities to each team member.
Week 5-6	Designing system architecture, data flow diagrams, fusion strategies, and model pipeline.
Week 7-8	Dashboard development and visualization
Week 8-9	Feature extraction from text and audio data
Week 9-10	Final system takes raw audio and outputs labeled transcripts.
Week 10-12	Preparing final report, formatting datasets, authoring research contributions.

1.6.2. Rationale behind Selected Methodology

The Agile methodology is selected because it allows for iterative development and continuous testing, which is essential for machine learning projects where model performance needs to be evaluated and refined continuously. The methodology supports rapid prototyping, early detection of issues, and flexibility to adapt to changing requirements. Additionally, regular sprint reviews enable continuous feedback and improvement throughout the development process.

Chapter 2: Problem Definition

This chapter discusses the precise problem to be solved, the objectives of the project, its scope, and the proposed solution. It outlines the system architecture, deliverables, and operating environment for the sentiment analysis system.

2.1. Problem Statement

Call centers generate massive amounts of conversational data every day, yet most organizations lack affordable and robust systems that can truly understand not only what customers say but also how they say it.Existing solutions are often restricted to text-based sentiment classification, ignoring important paralinguistic cues such as tone, pitch, stress, pauses, and emotional intensity, all of which carry crucial information about customer intent.

Moreover, enterprise-grade tools that provide partial multimodal analysis are often prohibitively expensive, limiting accessibility for academic research and small to medium-sized businesses. Current systems also fall short in capturing the natural dynamics of a conversation, such as interruptions, hesitation, sentiment drift, and agent responsiveness, resulting in incomplete insights.

More critically, these tools are largely descriptive and fail to provide predictive analytics, such as estimating the probability of a successful sale, which is a key metric for call centers aiming to optimize operations. There is a clear need for an open-source, affordable, and comprehensive solution that addresses these limitations.

Objectives

The primary objective is to design a low-cost, open-source, and student-friendly AI system capable of:

1. Transcribing telephonic conversations using state-of-the-art Automatic Speech Recognition (ASR)
2. Identifying and classifying customer sentiments and emotional states from both text and audio
3. Modeling conversational dynamics including interruptions, hesitation, and sentiment drift
4. Predicting the probability of successful sales conversions with interpretable AI techniques
5. Providing visual, interactive dashboards for managers to assess customer satisfaction and agent performance
6. Offering an extensible platform that can be integrated into existing CRM systems

2.2. Product Functions

The proposed system delivers a comprehensive suite of analytical capabilities spanning audio processing, multimodal feature extraction, machine learning inference, and interactive visualization. The following table enumerates core system functions

Table 3 For Product Functions

Function ID	Function Name	Description
PF-01	Audio Upload and Processing	Allows the user to upload an audio file in supported formats (.wav, .mp3) for diarization.
PF-02	Automatic Transcription	Convert speech to text using Whisper ASR
PF-03	Speaker Diarization:	Identify and separate customer and agent speech using Pyannote.audio
PF-04	Sentiment Analysis:	Analyze textual content for sentiment polarity and intensity
PF-05	Emotion Recognition:	Detect emotions from acoustic features such as pitch, tone, and MFCCs
PF-06	Sales Prediction:	Estimate conversion probability using ensemble learning models
PF-07	Dashboard Visualization	Display insights through interactive charts and graphs
PF-08	Export Results	Generate detailed reports on call analysis and agent performance

Solution

Our solution is a multimodal AI system that integrates multiple state-of-the-art technologies:

Whisper ASR: OpenAI's Whisper model for robust automatic speech recognition and transcription

Pyannote.audio: Neural building blocks for speaker diarization to separate customer and agent speech

DistilBERT/FinBERT: Transformer-based models for textual sentiment analysis with domain-specific fine-tuning

CNN+LSTM Architecture: Convolutional Neural Networks combined with Long Short-Term Memory networks for acoustic emotion recognition from speech features

XGBoost/LSTM Ensemble: Ensemble learning approach combining gradient boosting and recurrent neural networks for sales conversion prediction

Feature Fusion: Integration of textual and acoustic features for comprehensive multimodal analysis

Advantages of the Proposed System:

- Multimodal analysis combining audio and text for higher accuracy
- Real-time sales conversion probability prediction
- Open-source, cost-effective, and extensible design
- Explainable AI techniques for transparency and interpretability
- Potential integration with existing CRM systems using Agentic Workflows
- Capability for future extension to real-time analysis
- Captures hesitation cues and conversational dynamics

2.3. Proposed Architecture

The system architecture follows a modular, layered design enabling independent development, testing, and maintenance of subsystems while facilitating seamless integration through well-defined interfaces.

2.3.1 Architectural Overview

The overall architecture consists of the following major components:

Stage	Description
1. Data Ingestion Layer	Handles audio file uploads and initial processing.
2. Preprocessing Layer	Audio normalization, noise reduction, and format conversion
3. Feature Extraction Layer	Speech-to-text transcription using Whisper ASR Speaker diarization using Pyannote.audio Acoustic feature extraction (MFCCs, pitch, tone, energy) Text preprocessing and tokenization
4. Analysis Layer	Sentiment analysis using DistilBERT/FinBERT Emotion recognition using CNN+LSTM Conversational dynamics modeling
5. Prediction Layer	Sales conversion probability estimation using XGBoost/LSTM ensemble
6. Backend (Dot Product + Sigmoid)	The refined attractors are compared with frame embeddings using dot products. A sigmoid function generates final probabilities of which speaker is active at each frame.
7. Presentation Layer	Interactive dashboards built with React for visualization and insights
8. Data Storage Layer	PostgreSQL for structured data and MongoDB for unstructured data

2.3.2 Proposed Workflow

The end-to-end analysis workflow proceeds through the following stages:

Below is a workflow figure

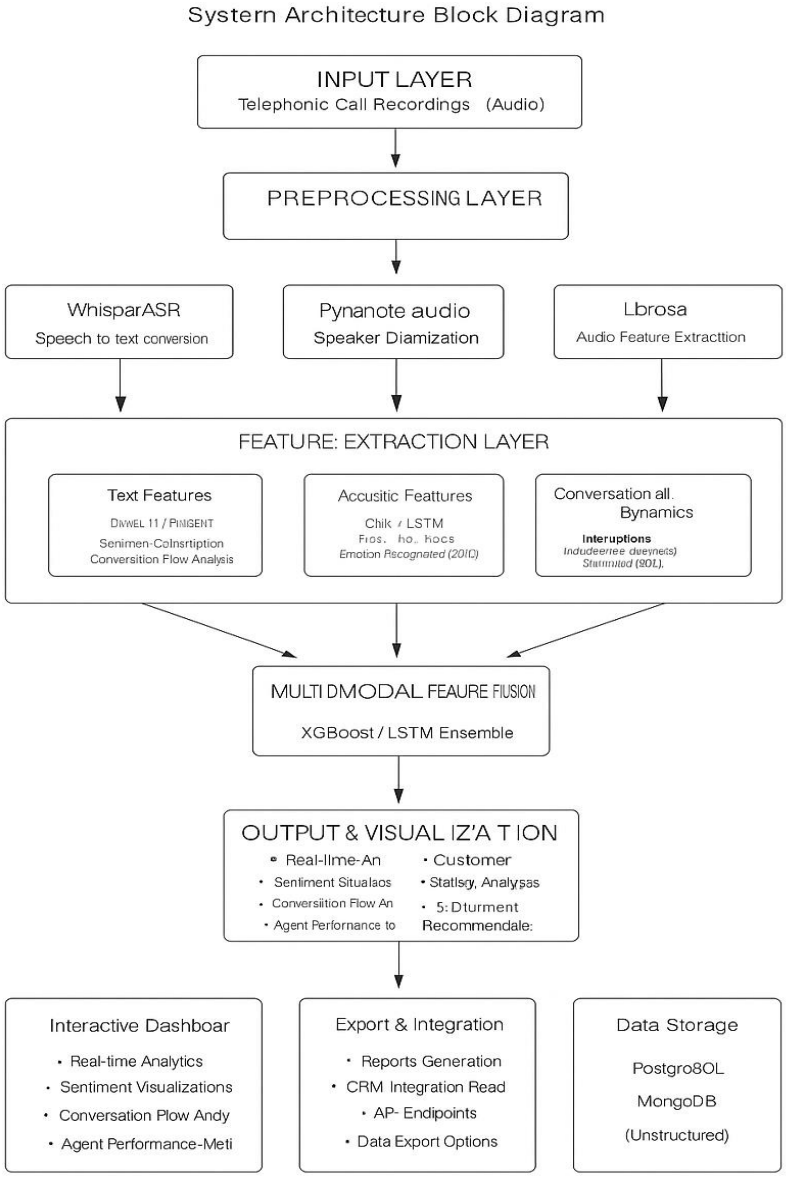

Figure 1 Proposed Workflow

2.4. Project Deliverables

The major outcomes and artifacts expected from this project are listed below:

Table 4 For Project Deliverables

Deliverable	Description
D1: Literature Review Report	Comprehensive survey of sentiment analysis, speech emotion recognition, multimodal fusion, and call center analytics research; comparative analysis of existing systems; identification of research gaps
D2: Whisper ASR Integration Module	Python module implementing Whisper-based speech-to-text transcription with timestamp generation; handles multiple audio formats; configurable model size (tiny/base/small/medium/large)
D3: Speaker Diarization Module	Python module implementing Pyannote.audio-based speaker segmentation; separates Customer vs. Agent speech; detects overlapping speech
D4: React Dashboard Fronten	Interactive web application providing real-time visualization of sentiment, emotion, conversion predictions; responsive design; Material-UI components
D5: PostgreSQL Database Schema	Relational database schema for conversation metadata, sentiment scores, emotion labels, predictions, user accounts; includes indexes and constraints
D6: System Integration Test	Comprehensive test suite covering unit tests (individual modules), integration tests (pipeline end-to-end), performance tests (latency, throughput); achieves ≥90% code coverage
D7: Technical Documentation	Detailed technical specifications: architecture diagrams, API documentation (OpenAPI/Swagger), model architectures, training procedures, hyperparameters; facilitates reproducibility

2.5. Operating Environment

The system's development, deployment, and execution require specific hardware and software configurations to ensure optimal performance.

Following are different types of requirements:

2.5.1 Hardware Environment

Table 5: Hardware Environment

Component	Minimum Requirement	Recommended Requirement
Processor (CPU)	Intel Core i5 / Ryzen 5	Intel Core i7 / Ryzen 7 or higher
GPU (for model training/inference)	NVIDIA GPU with 4GB VRAM (e.g., GTX 1650)	NVIDIA RTX 3060 or higher with ≥ 8GB VRAM and CUDA support
RAM	8 GB	16 GB or more
Storage	10 GB of free space (datasets + models)	50 GB SSD or higher for faster I/O
Audio Support	Capable of handling .wav/.mp3 input files	Same

2.5.2 Software Environment

Table 6 Software Environment

Software	Version / Requirement
Operating System	Windows 10/11, Ubuntu 20.04+, or macOS
Programming Language	Python 3.8 or above
Deep Learning Framework	PyTorch (with CUDA Toolkit if GPU available)
Python Libraries	NumPy, SciPy, Librosa, Scikit-learn, Pyannote.audio, Matplotlib, Pandas
Clustering Tool	Scikit-learn for k-means clustering
Model Dependencies	Transformer models (PyTorch), k-means, Sigmoid backend
Version Control	Git and GitHub for source code management

Chapter 3: Requirement Analysis

This chapter presents the Software Requirements Specification (SRS) for the multimodal sentiment analysis system. It includes use case models illustrating system interactions from user perspectives, comprehensive functional requirements defining system behaviors, and non-functional requirements specifying quality attributes, performance criteria, and design constraints.

3.1. Use Case Models

3.1.1. Use Case Diagram:

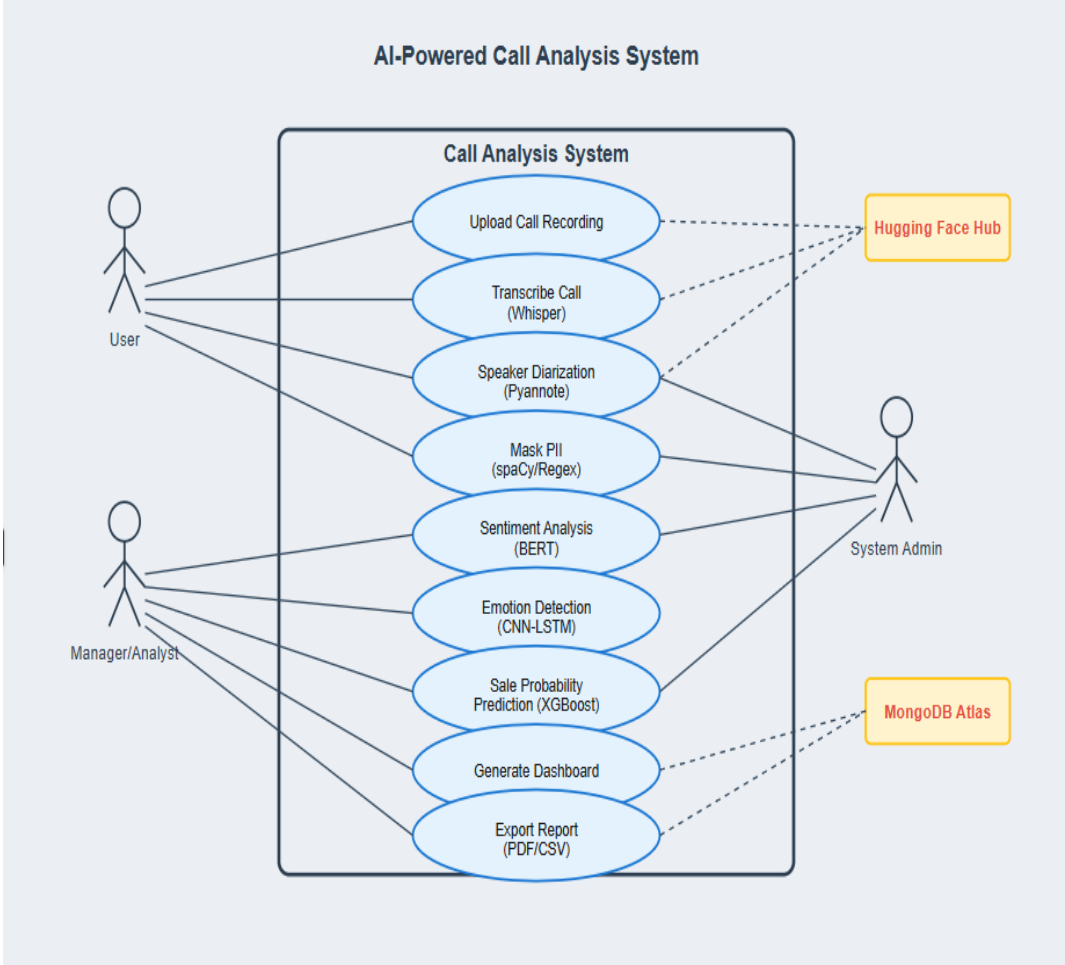

Figure 2 Use Case Diagram

3.1.2. Use Cases: Upload and Analyze Call Recording

Following are some use cases:

Table 7 Use Case-01

Use Case ID:	UC-01
Use Case Name:	Upload and Analyze Call Recording
Actors:	Call Center Manager, Sales Analyst
Description:	User uploads a call recording and the system performs comprehensive analysis including transcription, sentiment analysis, emotion recognition, and conversion prediction
Trigger:	User navigates to upload page and selects "Upload Call Recording" action
Preconditions:	1. User is authenticated and logged into the system 2. Audio file is in supported format (WAV, MP3, M4A) 3. System has sufficient storage and processing capacity
Postconditions:	1. Audio file is stored in the database 2. Complete analysis results are available 3. Dashboard displays insights and visualizations
Normal Flow:	1. User navigates to upload page 2. User selects audio file from local storage 3. System validates file format and size 4. User clicks "Upload and Analyze" button 5. System uploads file to server 6. System transcribes audio using Whisper ASR 7. System performs speaker diarization 8. System analyzes sentiment from transcript 9. System extracts acoustic features and recognizes emotions 10. System predicts sales conversion probability 11. System displays results on dashboard

	12. User reviews analysis results
Alternative Flows:	**Alt 1:** Invalid file format System detects unsupported format System displays error message User selects different file **Alt 2:** Processing failure System encounters error during analysis System logs error details
Exceptions:	**EX-01: System Overload** 1.Condition: Task queue has >50 pending jobs 2.Response: System displays "Server busy. Your analysis is queued (position #X). Estimated wait: Y minutes." 3.Recovery: Job processed when queue capacity available **EX-02: Model Loading Failure** 1.Condition: 1.Whisper/DistilBERT/CNN-LSTM models fail to load (file corruption, insufficient memory) 2.Response: System logs critical error; displays "Internal error: Unable to load analysis models. Contact administrator." 3.Recovery: Administrator restarts services; users retry after notification
Includes:	UC-04: Validate Audio File UC-05: Perform ASR & Diarization UC-07: Extract Multimodal Features< UC-08: Predict Sales Conversion UC-10: Generate Explainability Insights

Special Requirements:	**1.**Analysis completion time 2. audio duration (e.g., 10-minute call analyzed in ≤20 minutes) 3. Upload supports files up to 100MB with resumable transfer 4. Audio files encrypted at rest (AES-256) 5.User authentication required via JWT tokens 6.Role-based access control enforced 7.**Usability** Progress bar shows analysis stages 8. Estimated completion time displayed9. Mobile-responsive upload interface
Assumptions:	1.User has legal authorization to record and analyze conversations 2/Audio contains genuine spoken conversation (not music/ambient noise) 3.Recording quality sufficient for ASR (≥8kHz, SNR ≥10dB preferred) 4. Internet connection stable during upload (≥1Mbps upload speed) 5. Browser supports HTML5 file upload (Chrome 90+, Firefox 88+, Safari 14+).
Notes and Issues:	**Open Issue 1:** Should system support batch upload (multiple files simultaneously **Resolution:** Deferred to future release; current version processes one file at a time per user **Open Issue 2:** Maximum practical audio duration before memory constraints **Resolution:** Testing shows 60-minute recordings feasible on 16GB RAM; longer recordings may require chunking **Design Decision:** Asynchronous processing (Celery queue) chosen over synchronous to prevent HTTP request timeout and enable concurrent analysis

3.1.3. Use Case: View Performance Dashboard

Table 8 Use Case-02

Use Case ID:	UC-02
Use Case Name:	View Performance Dashboard
Actors:	Call Center Manager (Primary), Sales Analyst, System Administrator
Description:	User views interactive dashboard displaying aggregated metrics, trends, and insights from analyzed conversations
Trigger:	User navigates to dashboard URL or clicks "Dashboard" in navigation menu
Preconditions:	1. User is authenticated 2. At least one call recording has been analyzed 3. Dashboard data is up to date
Postconditions:	1. User has viewed relevant metrics and insights 2. User activity is logged
Normal Flow:	1. User navigates to dashboard page 2. System loads and displays key metrics 3. System shows sentiment trend charts 4. System displays emotion distribution graphs 5. System presents conversion rate statistics 6. User applies filters (date range, agent, outcome) 7. System updates visualizations based on filters 8. User drills down into specific conversations 9. System displays detailed analysis for selected call
Alternative Flows:	1. User uploads audio in unsupported format. 2. System displays an error: “Unsupported Audio Format (.aac, .flac, etc.). Only .wav or .mp3 are allowed.” 3. User is prompted to upload a compatible file. 4. Use case resets to Step 1.

Exceptions:	**1.**Query returns zero conversations for selected filters 2.System displays message: "No conversations match your filters. Try adjusting date range or criteria." 3.Charts show empty state illustrations with helpful hints 4. User adjusts filters → Return to Step 7
Includes:	None
Special Requirements:	**1.Performance:** Initial dashboard load Filter application and chart updates ≤500msSupport 50+ concurrent dashboard users **2.Usability:** Responsive design supporting desktop (1920×1080+), tablet (768×1024+), and mobile (375×667+) Color-blind friendly palette (use patterns in addition to colors) Keyboard navigation support (tab through filters, arrow keys for chart elements) **3.Accessibility:** WCAG 2.1 AA compliance Screen reader compatible with ARIA labels High contrast mode available
Assumptions:	1. User has stable internet connection (≥512kbps) for dashboard interactions 2. Browser supports modern JavaScript (ES6+), Canvas/SVG for visualizations 3. User understands basic data visualization concepts (line charts, pie charts, scatter plots) 4. Aggregate statistics precomputed and cached for performance (materialized views in PostgreSQL)
Notes and Issues:	**1.Design Decision:** Client-side rendering (React + D3.js) chosen over server-side for interactivity; initial data fetched via REST API, then cached in Redux store **2.Performance Optimization:** Implemented debouncing on filter changes (300ms delay) to reduce unnecessary API calls. 3.**Future Enhancement:** Real-time WebSocket updates for live call monitoring

3.1.4. Use Case: Generate Analysis Report

Table 9 Use Case 03 Retrain Speaker Embedding Model

Use Case ID:	UC-03
Use Case Name:	Generate Analysis Report
Actors:	Call Center Manager, Sales Analyst (Primary); System (Secondary)
Description:	User generates a comprehensive report containing analysis results, transcripts, and insights for one or multiple conversations
Trigger:	User selects conversations and clicks "Generate Report" button in dashboard or conversation details page
Preconditions:	1. User is authenticated 2. Conversations have been analyzed User has permission to generate reports
Postconditions:	1. Report is generated in PDF format 2. Report is available for download 3. Report generation is logged
Normal Flow:	1. User selects conversations for report 2. User chooses report template 3. User specifies report parameters 4. User clicks "Generate Report" button 5. System compiles analysis data 6. System formats report according to template 7. System generates PDF document 8. System provides download link 9.User downloads report
Alternative Flows:	**1.No Conversations Selected** 2.User clicks "Generate Report" without selecting conversations 3.System displays validation error: "Please select at least one conversation

	4.Generate button remains disabled until selection made
Exceptions:	**1.PDF Generation Failure** Condition: 2.ReportLab library error during rendering (e.g., memory exhaustion for 50+ conversation report) 3.Response: System logs error; displays "Report generation failed. Try reducing number of conversations or contact support." 4.Recovery: User selects fewer conversations or administrator increases memory allocation
Includes:	None
Special Requirements:	1.Performance:Single conversation report generation ≤10 seconds 2. Multi-conversation report (up to 20 conversations) ≤60 seconds 3.PDF file size optimized (charts as compressed images, <5MB per report) 4.Quality: Charts rendered at 300 DPI for print quality 5.Transcript formatting preserves readability (12pt font, adequate line spacing) 6.Professional appearance suitable for executive stakeholders 7.Storage: Temporary files auto-deleted after 24 hours to conserve disk space 8.User can regenerate reports anytime from stored analysis data |
Assumptions:	1.User has appropriate permissions to view sensitive conversation data 2. Company branding assets (logo, color scheme) configured by administrator 3. User's browser supports PDF download (all modern browsers) 4. Generated reports comply with data retention policies 5. User understands data visualizations and statistical metrics presented
Notes and Issues:	1.Design Decision: PDF chosen over DOCX for consistent cross-platform rendering and to prevent editing of official analytical reports 2.security Consideration: Report download links expire after 24 hours and require authentication to prevent unauthorized acces 3.Future Enhancement: Interactive HTML reports with embedded JavaScript charts for deeper exploration; scheduled automated reporting for managers

3.2. Functional Requirements

The functional requirements define the specific behaviors and functions of the system:

FR1: Audio Upload and Management

- FR1.1: The system shall allow users to upload audio files in WAV, MP3, and M4A formats
- FR1.2: The system shall validate audio file format and size before processing
- FR1.3: The system shall store uploaded audio files securely in the database
- FR1.4: The system shall allow users to view a list of previously uploaded recordings

FR2: Speech Recognition and Transcription

- FR2.1: The system shall transcribe audio recordings to text using Whisper ASR
- FR2.2: The system shall achieve minimum 85% transcription accuracy for clear audio
- FR2.3: The system shall identify and separate speaker segments using diarization
- FR2.4: The system shall label speakers as "Customer" and "Agent"

FR3: Sentiment Analysis

- FR3.1: The system shall analyze transcribed text for sentiment polarity (positive, negative, neutral)
- FR3.2: The system shall calculate sentiment scores for each conversation segment
- FR3.3: The system shall track sentiment changes throughout the conversation
- FR3.4: The system shall identify key phrases contributing to sentiment

FR4: Emotion Recognition

- FR4.1: The system shall extract acoustic features (MFCCs, pitch, tone) from audio
- FR4.2: The system shall classify emotions into categories (happy, sad, angry, neutral, frustrated)
- FR4.3: The system shall provide emotion confidence scores
- FR4.4: The system shall detect emotion transitions during conversations

FR5: Conversational Dynamics Analysis

- FR5.1: The system shall detect interruptions and overlapping speech
- FR5.2: The system shall identify hesitation patterns (pauses, filler words)
- FR5.3: The system shall measure speaking time ratio between customer and agent
- FR5.4: The system shall analyze conversation flow and turn-taking patterns

FR6: Sales Conversion Prediction

- FR6.1: The system shall predict sales conversion probability as a percentage
- FR6.2: The system shall identify key factors influencing conversion prediction
- FR6.3: The system shall provide confidence intervals for predictions
- FR6.4: The system shall update predictions based on conversation progress

FR7: Dashboard and Visualization

- FR7.1: The system shall display sentiment trends over time in graphical format
- FR7.2: The system shall show emotion distribution charts
- FR7.3: The system shall present conversion probability with visual indicators
- FR7.4: The system shall provide interactive filtering and drill-down capabilities
- FR7.5: The system shall display key metrics and statistics summary

3.3. Non-Functional Requirements

Non-functional requirements describe **how well the system should perform**.

3.3.1. Usability

- The user interface shall be intuitive and require minimal training
- The system shall provide clear error messages and guidance
- The dashboard shall be accessible on desktop and tablet devices
- The system shall support keyboard navigation and accessibility features

3.3.2. Reliability

- The system shall have 95% uptime availability
- The system shall handle errors gracefully without data loss

- The system shall implement automatic backup mechanisms
- The system shall recover from failures within 5 minutes

3.3.3. Performance

- The system shall process a 10-minute audio file within 3 minutes
- The dashboard shall load within 2 seconds
- The system shall support concurrent processing of up to 5 audio files
- Database queries shall return results within 1 second

3.3.4. Supportability

- The system shall be modular to facilitate maintenance and updates
- The code shall follow PEP 8 style guidelines for Python
- The system shall include comprehensive logging for debugging
- The system shall provide API documentation using OpenAPI/Swagger

3.3.5. Design Constraints

- The system shall be developed using Python 3.10 or higher
- The frontend shall be built using React 18
- The system shall use PostgreSQL for relational data storage
- The system shall use MongoDB for unstructured data storage

3.3.6. Licensing Requirements

- The system shall use only open-source libraries with permissive licenses
- The system shall be released under MIT or Apache 2.0 license
- All third-party dependencies shall be properly attributed

References:

1. IBM Corporation. (2025). *Watson Tone Analyzer*. Retrieved from https://www.ibm.com/watson/services/tone-analyzer/
2. Amazon Web Services. (2025). *Amazon Connect Contact Lens*. Retrieved from https://aws.amazon.com/connect/contact-lens/
3. Google Cloud. (2025). *Cloud Speech-to-Text and Natural Language APIs*. Retrieved from https://cloud.google.com/
4. Radford, A., Kim, J. W., Xu, T., Brockman, G., McLeavey, C., & Sutskever, I. (2023). Robust speech recognition via large-scale weak supervision. In *Proceedings of the 40th International Conference on Machine Learning* (ICML 2023), pp. 28492-28518.
5. Bredin, H., Yin, R., Coria, J. M., Gelly, G., Korshunov, P., Lavechin, M., Fustes, D., Titeux, H., Bouaziz, W., & Gill, M. P. (2025). *Pyannote.audio: Neural building blocks for speaker diarization*. Retrieved from https://github.com/pyannote/pyannote-audio
6. Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.
7. Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (KDD), pp. 785-794. https://doi.org/10.1145/2939672.2939785
8. Zhao, J., Mao, X., & Chen, L. (2019). Speech emotion recognition using deep 1D & 2D CNN LSTM networks. *Biomedical Signal Processing and Control*, 47, 312-323.
9. Poria, S., Cambria, E., Bajpai, R., & Hussain, A. (2017). A review of affective computing: From unimodal analysis to multimodal fusion. *Information Fusion*, 37, 98-125.
10. Zadeh, A., Liang, P. P., Mazumder, N., Poria, S., Cambria, E., & Morency, L. P. (2018). Memory fusion network for multi-view sequential learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).